

# Efficient Three-Dimensional Capacitance Calculation

by

Keith Shelton Nabors

B.E.S., Johns Hopkins University (1983)  
S.M., Massachusetts Institute of Technology (1990)

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

May 1993

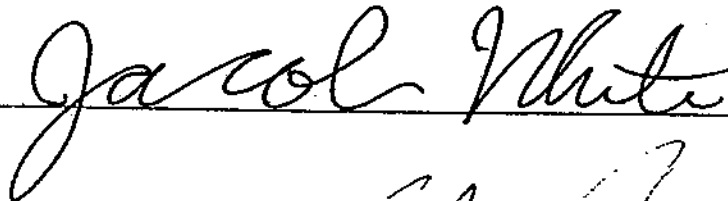
© Massachusetts Institute of Technology 1993

Signature of Author



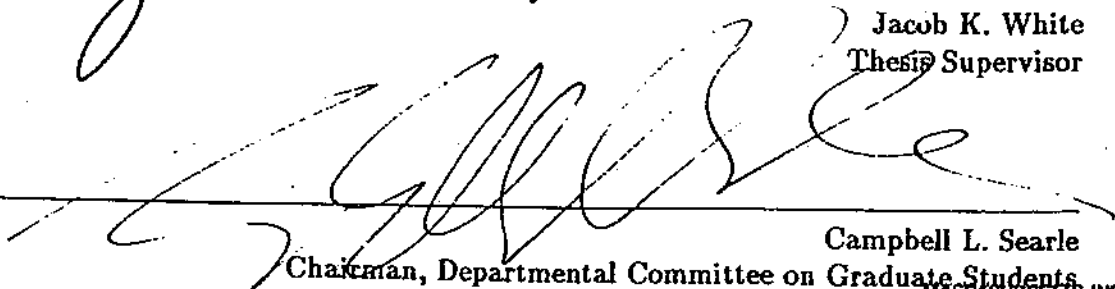
Department of Electrical Engineering and Computer Science  
May 14, 1993

Certified by



Jacob K. White  
Thesis Supervisor

Accepted by



Campbell L. Searle  
Chairman, Departmental Committee on Graduate Students

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

MAY 14 1993



# Efficient Three-Dimensional Capacitance Calculation

by

Keith Shelton Nabors

Submitted to the Department of Electrical Engineering and Computer Science  
on May 14, 1993, in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy

## Abstract

This thesis describes a method for the efficient calculation of the capacitances of general three-dimensional conductor systems. The method differs from other techniques primarily in its greatly reduced algorithmic complexity. The linear system arising from the discretization of the associated potential problem is solved using a rapidly convergent iterative method with the iterates calculated using a fast multipole algorithm. For a system of  $m$  conductors leading to an  $n \times n$  linear system, the method requires computer time that scales nearly as  $mn$  and storage nearly proportional to  $n$ . In the conventional approach computer time scales as  $n^3$  and storage as  $n^2$ . The resulting savings in computer time and memory allow accurate three-dimensional capacitance calculations to be integrated into an iterative design process.

Thesis Supervisor: Jacob K. White

Associate Professor of Electrical Engineering and Computer Science



---

---

# Acknowledgments

Although this thesis bears my name, it is more the result of my interactions with other people than of my own thinking or intuition. I have gradually come to realize that any knowledge I might possess can only act as a catalyst for the interchange and advancement of ideas that is research. My colleagues were more than just helpful—they were an essential part of the process.

The bulk of this thesis is the result of a joint effort with Prof. Jacob White. I had thought such a thing impossible because of the distance a graduate student puts between himself and his advisor as a sign of respect. Working with Jacob has taught me that fear is the only motivation for that distancing, and having the courage to put it aside makes real research possible, and fosters a stronger sort of respect. Without Jacob's inspiration I might have never learned this important lesson.

The members of my research group have also been an essential part of this work. Don Baltus has acted as a general sounding board since the beginning, and has helped considerably, particularly with computer science issues. Khalid Rahmat has also been willing to help me think about almost anything and has been especially helpful in developing intuition about electrostatics problems. Other notable collaborators include Songmin Kim, Ricardo Telichevsky and Miguel Silveira who were involved with the details of the implementation of the algorithms developed in this thesis.

Several people outside the group have also contributed. Dr. Albert Ruehli and Dr. David Ling of IBM described their earlier work in capacitance calculation, and Dr. Steven McCormick of the Digital Equipment Corporation explained aspects of DEC's capacitance calculation software. This information was the starting point for the work described in the thesis. Also Prof. Andrew Lumsdaine of Notre Dame University pointed out some relevant research that had a impact on the early directions of the effort.

One of the most rewarding aspects of this work has been producing a capacitance calculation program that is actually useful. The program's users have been an essential part of the capacitance calculation algorithm's development. The members of Prof. Stephen Senturia's MEMCAD group, especially Brian Johnson, have gone out of their way to demonstrate the need for several key improvements. Other users that have contributed significantly include Joe Kung of Motorola, Kenji Matsuoka of SUNY Stony Brook, Tze-Ting Fang of Ansoft and Anyu Lee

of Amdahl. Mark Bales, until recently with Cadence Design Systems, arranged for me to begin integrating the program into Cadence's CAD system. The Cadence system was a particularly challenging user that led to several new lines of thinking. Dr. F. T. Korsmeyer and Prof. J. N. Newman of the Ocean Engineering Department developed more general algorithms using the program as a starting point, giving a different perspective on the basic algorithm. Prof. Newman also took the time to serve on my thesis committee.

In addition to those already mentioned, I have received noteworthy encouragement from long-time friends David Bowen, Chris Berry, Ken Chou and Wenken Ling. Also the emotional support of my family and Michele Popper has been nothing short of essential. Michele also made several of the figures, and helped with the typing and proof reading.

I have followed a particularly crooked path to the M.I.T. Ph.D., one that seemed to rarely follow Departmental guidelines. I did virtually everything late, in the wrong order, and without the proper documentation. Fortunately the Department is not some faceless bureaucracy to graduate students—it is instead the sainted Marilyn Pierce. Countless times I have gone into her office to ask about some horrible thing that didn't seem to fall under her jurisdiction only to have her say, "Oh, that...don't worry, I'll fix it." If it weren't for her and her connections at the Registrar's office my life would have been considerably more painful. My other link to Departmental guidelines was through my academic advisor, Prof. George Verghese. Although he often told me to ask Marilyn when it came to actual guidelines, his encouragement and genuine concern helped me through some difficult times, and his gentle prodding helped me finally get this done. Finally, Prof. Steven Leeb was kind enough to help me fulfill one of the last Departmental requirements by becoming a member of my thesis committee at the very last minute.







---

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>17</b>
<b>2</b>	<b>Approaches to the Capacitance Problem</b>	<b>19</b>
2.1	Mathematical Formulation of the Capacitance Problem . . . . .	19
2.1.1	Indirect Formulations . . . . .	20
2.1.2	Capacitance Calculation with Charge Layers . . . . .	23
2.2	Numerical Treatment of the Capacitance Problem . . . . .	24
2.2.1	The Method of Weighted Residuals . . . . .	24
2.2.2	Stationary-Functional and Finite-Element Methods . . . . .	27
2.2.3	Finite-Difference Methods . . . . .	27
2.3	Previous Work . . . . .	28
2.3.1	Methods for Special Geometries . . . . .	28
2.3.2	Methods for General Three-Dimensional Geometries . . . . .	28
<b>3</b>	<b>Multipole Accelerated Capacitance Calculation</b>	<b>31</b>
3.1	First-Kind Integral-Equation Discretization . . . . .	31
3.1.1	Weighted-Residuals Discretization . . . . .	31
3.1.2	Discretization Build and Solve Costs . . . . .	33
3.2	Accelerating the Discretized System Solution . . . . .	33
3.2.1	Solving the Discretized System Iteratively . . . . .	34
3.2.2	Efficient Matrix-Vector Products with the Multipole Algorithm . . . . .	37
3.2.3	Maintaining Efficiency with the Adaptive Multipole Algorithm . . . . .	45
3.2.4	Preconditioning the Multipole Accelerated Iterative Method . . . . .	54
3.3	Extending the Algorithm to Problems with Multiple Dielectrics . . . . .	65
3.3.1	Changing the Mathematical Formulation . . . . .	65
3.3.2	Weighted-Residuals Discretization . . . . .	66
3.3.3	Multipole Accelerated Iterative Solution . . . . .	69
3.3.4	Preconditioning . . . . .	73
3.3.5	Capacitance Matrix Calculation from the Charge Layer Density . . . . .	76

<b>4</b>	<b>Multipole Accelerated Capacitance Calculation Algorithm Evaluation</b>	<b>77</b>
4.1	Implementation . . . . .	77
4.1.1	Capacitance Calculation Algorithm . . . . .	77
4.1.2	Multipole Algorithm Initialization . . . . .	80
4.1.3	Multipole Algorithm Field Calculation . . . . .	87
4.2	Convergence and Error . . . . .	88
4.2.1	Uniqueness and Well-Posedness . . . . .	89
4.2.2	Weighted-Residuals Discretization Convergence and Error . . . . .	90
4.2.3	Linear System Solution Convergence and Error . . . . .	90
4.3	Algorithmic Complexity . . . . .	96
4.3.1	Parent-Child Translation Operation Complexity . . . . .	99
4.3.2	Interaction-Operation Complexity . . . . .	101
4.4	Performance . . . . .	104
4.4.1	Accuracy . . . . .	104
4.4.2	Execution Time . . . . .	118
4.4.3	Memory Use . . . . .	129
<b>5</b>	<b>Conclusion and Future Work</b>	<b>135</b>
<b>A</b>	<b>Surface Layers</b>	<b>139</b>
A.1	Charge Layers . . . . .	139
A.2	Dipole Layers . . . . .	142
<b>B</b>	<b>Multipole Algorithm Formulas</b>	<b>145</b>
B.1	Formula Components . . . . .	145
B.1.1	Real Valued Expansion Coefficients . . . . .	145
B.1.2	Spherical Harmonics . . . . .	146
B.2	Real Coefficient Multipole Algorithm Formulas . . . . .	146
B.2.1	Multipole Expansion (Q2M, M2P) . . . . .	147
B.2.2	Local Expansion (Q2L, L2P) . . . . .	147
B.2.3	Multipole-Expansion Shift (M2M) . . . . .	148
B.2.4	Multipole to Local Expansion Conversion (M2L) . . . . .	149
B.2.5	Local Expansion Shift (L2L) . . . . .	150
B.2.6	Electric Fields From Multipole Expansions (M2E) . . . . .	150
B.2.7	Electric Fields From Local Expansions (L2E) . . . . .	152
<b>C</b>	<b>Galerkin Panel Integral Approximation</b>	<b>155</b>
<b>D</b>	<b>Local Expansion Efficiency</b>	<b>159</b>

---



---

## List of Figures

3-1	Evaluation of $d$ distant panel potentials at $d$ collocation points with the multipole expansion. . . . .	39
3-2	The combining of lower-level cube multipole expansions to form multipole expansions for their parent cubes. . . . .	40
3-3	Evaluation of the potential at $\mathbf{x}_i$ . . . . .	41
3-4	The conversion of several distant multipole expansions into one local expansion for evaluation at the local collocation points. . . . .	43
3-5	One step (a) in the construction of the local-expansion hierarchy consists of building many lower-level child cube expansions using the upper-level parent expansions and distant multipole expansions as in (b). . . . .	44
3-6	The basic operations of the fast multipole algorithm. . . . .	46
3-7	The multipole-expansion hierarchies constructed by the non-adaptive (a) and adaptive (b) multipole algorithms for an example panel-charge distribution. . . . .	49
3-8	The local expansion hierarchies constructed by the non-adaptive (a) and adaptive (b) multipole algorithms for an example panel-charge distribution. . . . .	51
3-9	Partitionings used in Algorithm 3.3, (b), and by the adaptive algorithm of [19], (a). . . . .	55
3-10	Seven $1m \times 1m$ parallel panels spaced $1m$ apart. . . . .	58
3-11	GMRES polynomials for $Pq = f$ (a) and $\bar{C}Pq = \bar{C}f$ (b), with $P$ and $\bar{C}$ given by (3.28) and (3.31) respectively, and where $f$ is the first column of the identity matrix. . . . .	59
3-12	The steps leading to the third row of the preconditioner $\bar{C}$ for the example $P$ ("x" denotes a non-zero element). . . . .	61
3-13	A one-dimensional cubic partitioning of a region containing panels. . . . .	63
3-14	Dark areas indicate the position of those entries of $P$ used to form the preconditioner for the woven1 discretization of Figure 4-9. . . . .	64
3-15	The part of the normal electric field on the dielectric panel due to distant charge panels can be approximated with a divided difference. . . . .	72

4-1	The cube geometry leading to the minimum separation distance, $r_s - R_s$ , between charge and evaluation points for a cube hierarchy with $d \times d \times d$ lowest-level cubes.	95
4-2	A parent-child translation computation graph (left) and the corresponding 2-D panel distribution and expansion hierarchies.	100
4-3	An example clustered charge distribution. Dark squares indicate filled lowest-level cubes; the shaded region is A's neighbor cube set, and the remaining non-empty cubes are in A's interaction set.	102
4-4	The sphere (left) and coated sphere (right) discretizations. Some of the dielectric panels have been removed to expose the conductor panels in the coated sphere discretization.	105
4-5	The cube (left) and plate (right) discretizations.	106
4-6	The backplane connector example. The actual discretization used for the computation is much finer than the one illustrated.	107
4-7	The complete DRAM model, (a), and with the dielectric interfaces removed for clarity, (b).	108
4-8	The line10 discretization of a ten-meter line.	109
4-9	The $2 \times 2$ woven-bus problem: bars have $1m \times 1m$ cross sections. The three discretizations are obtained by replacing each square face in (a) with the corresponding set of panels in (b).	110
4-10	The three $2 \times 2$ coated woven-bus discretizations are obtained by replacing the two dielectric interfaces in (a) with the corresponding set of panels in (b). Conductor faces are discretized as in Figure 4-9. Part of one dielectric interface has been removed to expose conductor 1 for the illustration.	111
4-11	The analytic and calculated potential as a function of radial distance for the coated capacitor of Figure 4-12 with $\epsilon_{ratio} = 10$ for $b = 1.5$ (top), $b = 2.0$ (middle) and $b = 2.5$ (bottom).	115
4-12	A schematic drawing of the coated sphere problem and the corresponding equivalent charge representation used by FASTCAP.	116
4-13	The analytic and calculated potential as a function of radial distance for the coated capacitor of Figure 4-12 with $\epsilon_{ratio} = 10$ , $b = 2$ using 10560 panels with $o = 1$ (top), and 864 panels with $o = 7$ (bottom).	117
4-14	Execution time in DEC 3000AXP/500 CPU minutes as a function of $mn$ for AMA and explicit GMRES applied to the woven-bus problem, Figure 4-9, (a) and the coated woven-bus problem, Figure 4-10 (b).	119
4-15	Execution time in DEC 3000AXP/500 CPU minutes as a function of $mn$ for PAMA applied to the woven-bus problem, Figure 4-9, (a) and the coated woven-bus problem, Figure 4-10 (b).	122

4-16	A schematic illustration of the diaphragm problem. The two plates are $0.02\mu\text{m}$ apart at the center. . . . .	123
4-17	The $5 \times 5$ woven-bus discretization is obtained by replacing each face in the illustration with nine panels, as for the woven1 problem of Figure 4-9. . . . .	125
4-18	The sixteen preconditioner test problems leading to the data in Table 4-15 result from varying the distance between the cubes and the dielectric interface, $h$ , and using the different face discretizations in (b) for all the conductor and dielectric faces shown in (a) to vary the panel edge with ratio, $\alpha_{width}$ . . . . .	127
4-19	Eigenvalues of the effective system matrices used by AMA (a) and PAMA (b) for the Figure 4-18 problem with $h = 5 \times 10^{-4}\text{m}$ and $\alpha_{width} = 0.1$ . . . . .	130
4-20	Memory use in megabytes as a function of number of panels $n$ for PAMA and the full-matrix methods applied to the woven-bus problem, Figure 4-9, (a) and the coated woven-bus problem, Figure 4-i0 (b). . . . .	131
A-1	Taking the limit from the positive (a) and negative (b) sides of the surface. . . . .	140



---



---

## List of Tables

3-1	GMRES polynomials for $Pq = f$ (a) and $\bar{C}Pq = \bar{C}f$ (b), with $P$ and $\bar{C}$ given by (3.28) and (3.31) respectively, and where $f$ is the first column of the identity matrix. . . . .	58
4-1	Multipole accelerated capacitance-calculation algorithm (Algorithm 4.1) time (in multiply-adds) and storage cost (in double precision numbers) bounds. Input, output, allocation and assign times are taken as zero. . . . .	97
4-2	Adaptive multipole algorithm initialization (Algorithm 4.2) time (in multiply-adds) and storage cost (in double precision numbers) bounds. Allocation and assign times are taken as zero. Constants are defined in Table 4-1. . . . .	98
4-3	Capacitance values (in pF) illustrating FASTCAP's accuracy. †By analytic calculation. ‡From [109, 63]. ◊ From [109]. . . . .	105
4-4	DRAM Bit-line capacitance matrix, fF. . . . .	109
4-5	Capacitance values (in pF) comparing FASTCAP to the empirically matched formulas of [21]. . . . .	110
4-6	Capacitance values (in pF) illustrating FASTCAP's accuracy for the complicated homogeneous-dielectric geometry of Figure 4-9. . . . .	112
4-7	Capacitance values (in pF) illustrating FASTCAP's accuracy for the complicated multiple-dielectric geometry of Figure 4-10. . . . .	112
4-8	Errors in the largest and smallest capacitances in the woven1 discretization of Figure 4-9 illustrating the effect of expansion order and iterative loop tolerance on FASTCAP's accuracy for problems with a homogeneous dielectric. . . . .	113
4-9	Errors in the largest and smallest capacitances in the wovend1l discretization of Figure 4-10 illustrating the effect of expansion order and iterative loop tolerance on FASTCAP's accuracy for problems with multiple dielectrics. . . . .	113
4-10	Errors in the computed capacitance for the coated capacitor of Figure 4-12 with $b = 2$ illustrating the effect of $\epsilon_{ratio}$ and the dielectric-panel integral approximation order on FASTCAP's accuracy. . . . .	114

4-11	Errors in the capacitance computed using $o = 1$ and progressively finer discretizations for the coated capacitor of Figure 4-12 with $b = 2$ and $\epsilon_{ratio} = 10$ . . . . .	116
4-12	The total number of iterations required to solve for the $4 \times 4$ capacitance matrices of the woven bus of Figure 4-9 (a) and coated woven bus of Figure 4-10 (b) to give the data plotted in Figure 4-14. The parenthesis indicate an extrapolated value for a calculation that needed excessive memory. . . . .	120
4-13	The total number of iterations required to solve for the $4 \times 4$ capacitance matrices of the woven bus of Figure 4-9 (a) and coated woven bus of Figure 4-10 (b) to give the data plotted in Figure 4-15. The symbols † and ‡ indicate calculations that use $D = 5$ and $D = 3$ spatial partitioning levels respectively; all others use $D = 4$ . . . . .	121
4-14	The DEC 3000AXP/500 CPU minutes (a) and total number of iterations (b) required to solve for the capacitance matrices of a representative set of problems. Values in parenthesis are extrapolated. . . . .	124
4-15	The DEC 3000AXP/500 CPU seconds used by AMA (a) and PAMA (b) to calculate capacitances for the test problems of Figure 4-18, with the corresponding iteration counts (c) and (d). . . . .	128
4-16	The memory, in megabytes, required to solve for the capacitance matrices of a representative set of problems. The value in parenthesis is extrapolated. . . . .	132
C-1	Evaluation points and weights used in the approximation of (C.6). . . . .	157



---

---

# Introduction

The need for fast three-dimensional capacitance calculation is particularly strong in integrated-circuit and circuit-packaging design. Capacitances of complicated conductor geometries are used to estimate capacitive loading and coupling effects both on and off chip [108, 29]. Performance analyses of dynamic random access memory (DRAM) circuits [92, 68], GaAs metal-semiconductor field-effect transistor (MESFET) circuits [48] and integrated circuit package connectors [143, 142, 118], for example, hinge on the extraction of parasitic capacitances from conductor geometries that must be treated in three dimensions. In other cases, explicit capacitances that are part of an integrated-circuit design must be evaluated. These applications include binary-weighted-capacitor-array analog-to-digital converters [80, 124], microfabricated motors [81, 70] and various sensors that measure pressure and humidity with on-chip capacitive elements [64, 129, 25, 116]. Also, certain models of microstrip transmission line discontinuities are based on three-dimensional capacitances that must be calculated from conductor geometries. Such analyses typically assume only quasi-transverse-electromagnetic (quasi-TEM) mode excitation [120, 119, 121, 137] or include hybrid modes [104] and are computationally less involved alternatives to full-wave simulations for lower frequencies. Still other areas of engineering analysis that employ three-dimensional capacitance calculations are surface-acoustic-wave-transducer [39, 101], fluid-flow-sensor [2] and electric-machine [24, 102] design.

This thesis describes a method for the efficient calculation of the capacitances of general three-dimensional conductor systems. The method differs from other techniques primarily in its greatly reduced algorithmic complexity. The resulting savings in computer time and memory allow accurate three-dimensional capacitance calculations of complex structures to be integrated into an iterative design process.

The fast capacitance calculation algorithm described here is based on the work of Rokhlin [107] and Greengard [52, 54]. The approach involves solving the linear system arising from the discretization of an associated potential problem using a rapidly convergent iterative method, with the iterates calculated using a fast multipole algorithm. For a system of  $m$  conductors

leading to an  $n \times n$  linear system, the method requires computer time nearly proportional to  $mn$  and storage nearly proportional to  $n$ . For comparison, in the most commonly used approach computer time scales as  $n^3$  and storage as  $n^2$ .

Chapter 2 begins with a discussion of the capacitance-calculation potential problem and its common mathematical formulations. It then presents standard discretization techniques for the various formulations, and concludes with a description of the formulations and discretizations used in earlier work. Chapter 3 presents the major contribution of this thesis, a method for greatly accelerating the solution of the discretized problem. The chapter first describes the particular potential-problem formulation and discretization used in the thesis to calculate the capacitances associated with ideal conductors in a homogeneous dielectric. It then develops a preconditioned, multipole accelerated iterative method that solves the discretized problem using substantially less computer time and memory than the standard approaches. The chapter concludes with a description of how the acceleration method can be extended to problems with several regions, each with a different permittivity. Chapter 4 evaluates the algorithm developed in Chapter 3 by constructing an implementation of the algorithm and examining the implementation's convergence and error properties, algorithmic complexity, and performance in actual capacitance calculations. Four appendices follow the conclusion and future work chapter. Appendix A presents useful facts about surface-layer potentials, the details of the multipole algorithm transformations are given in Appendix B, Appendix C examines details of the adaptive multipole algorithm and Appendix D gives the two-dimensional, integral-approximation formulas used for problems with high permittivity regions.

Much of the work presented in this thesis exists in published form. The basic multipole acceleration technique (Sections 3.2.1, and 3.2.2) is described in [88, 87], the adaptive (Section 3.2.3) and preconditioning (Section 3.2.4) enhancements are presented in [83, 84], extensions of the basic algorithm to problems with multiple dielectrics (Section 3.3) are treated in [91, 90, 89], and the method's algorithmic complexity (Section 4.3) and extensions to problems with different boundary conditions are examined in [85, 86].

---

---

# Approaches to the Capacitance Problem

A capacitance-calculation method begins with one of the mathematical formulations of the capacitance problem described in Section 2.1. Discretization using one of the techniques of Section 2.2 then yields equations which are solved for the capacitances. Section 2.3 classifies previous capacitance-calculation work in terms of their formulation and discretization methods.

## 2.1 Mathematical Formulation of the Capacitance Problem

Consider a system of  $m$  ideal conductors in free space<sup>1</sup>. The capacitance of such a structure can be summarized by an  $m \times m$  symmetric matrix  $C$ . The  $i$ -th row of  $C$  has a positive diagonal entry  $C_{ii}$ , representing the self-capacitance of conductor  $i$ , and negative off-diagonal entries  $C_{ij}$ , representing coupling between conductors  $i$  and  $j$ ,  $j = 1, 2, \dots, m, j \neq i$ .

The capacitance of an ideal two-terminal capacitor can be measured by connecting it to a one volt source and observing the charge on the positive plate. Since the capacitance is equal to the positive-plate charge divided by the capacitor voltage, the charge is numerically equal to the capacitance. The extension of this procedure to the many-conductor case amounts to raising one conductor to one volt, grounding the rest and calculating the charge on each conductor. With conductor  $j$  at one volt and the rest grounded,  $C_{ij}$  is the charge on conductor  $i$ ,  $i = 1, 2, \dots, m$ . Repeating the procedure  $m$  times gives the  $m$  columns of  $C$ .

The capacitance matrix calculation for  $m$  conductors is therefore equivalent to solving for the conductor charges in  $m$  potential problems. In these problems, the conductor potentials are specified and the potential must satisfy Laplace's equation in the region between the conductors. Calling the conductor surfaces  $S_c$  and the free space between them  $V$ , leads to the standard

---

<sup>1</sup>Section 3.3 extends the formulation so that it is applicable to conductors surrounded by arbitrary, piecewise-constant dielectric regions.

exterior Dirichlet problem formulation for the potential  $\psi$ ,

$$\nabla^2 \psi(x) = 0, \quad x \in V; \quad (2.1)$$

$$\psi(x) = f(x), \quad x \in S_c; \quad (2.2)$$

$$\lim_{\|x\|_2 \rightarrow \infty} \psi(x) = 0. \quad (2.3)$$

Here  $f$  fixes the conductor potentials, either one or zero, and  $\|\cdot\|_2$  is the Euclidean norm.

The full exterior Dirichlet problem solution is  $\psi$  throughout  $V$ . For capacitance calculations only the total induced charge on each conductor, given by normal derivatives of  $\psi$  on  $S_c$ , is required. The techniques of Section 2.1.1 are somewhat indirect since the charge totals are calculated by first calculating the electric field  $E = -\nabla\psi$  from the complete Dirichlet problem solution and then applying Gauss's Law to each conductor. The integral equation formulations of Section 2.1.2 are simpler since they relate the induced charge directly to the specified boundary potential  $f(x)$  without explicitly solving (2.1-2.3).

### 2.1.1 Indirect Formulations

This section describes how the exterior Dirichlet problem (2.1-2.3) can be formally solved by Green's function techniques, separation of variables, energy minimization or dipole-layer-density methods. Once the potential  $\psi$  throughout  $V$  is found using one of these methods, the total charge on conductor  $i$ ,  $Q_i$ , is calculated using Gauss's Law applied to any surface,  $W_i$ , enclosing conductor  $i$ ,

$$Q_i = \epsilon_0 \iint_{W_i} (-\nabla\psi \cdot n_i) dW_i.$$

Here  $n_i$  is an outward normal to the surface  $W_i$ . For the problem with conductor  $j$  at unit potential,  $Q_i$  is numerically equal to the capacitance matrix entry  $C_{ij}$ . Because the charge  $Q_i$  is calculated from the full exterior problem solution, rather than directly from the boundary conditions (2.2-2.3), the methods of this section are called indirect methods.

### Green's Function Techniques

The classical Green's function technique for the solution of (2.1-2.3) starts from Green's Theorem applied to the potential  $\psi$  and the appropriate Dirichlet Green's function,  $G_D(x, x')$  [61, 46, 153],

$$\psi(x) = \iiint_V \rho(x') G_D(x, x') dV' + \epsilon_0 \iint_{S_c} \left[ G_D(x, x') \frac{\partial \psi(x')}{\partial n'} - \psi(x') \frac{\partial G_D(x, x')}{\partial n'} \right] dS'_c.$$

Here  $\rho$  represents charges in the volume  $V$  and  $n'$  is an outward normal to the conductor surfaces.

By definition,  $G_D(x, x')$  is the potential at point  $x$  produced by a unit charge at point  $x'$  with all the conductors at zero potential. The Green's function includes the effects of charge

induced on the conductors and is therefore a function of the problem geometry. Since there is no volume charge  $\rho$  and  $G_D(x, x') = 0$  for  $x' \in S_c$ , Green's Theorem reduces to an expression for the potential in  $V$  in terms of the specified potentials,

$$\begin{aligned}\psi(x) &= -\epsilon_0 \iint_{S_c} \psi(x') \frac{\partial G_D(x, x')}{\partial n'} dS'_c \\ &= -\epsilon_0 \iint_{S_c} f(x') \frac{\partial G_D(x, x')}{\partial n'} dS'_c,\end{aligned}\tag{2.4}$$

where  $f$  is as in (2.2). In general, finding the geometry dependent  $G_D(x, x')$  is difficult, making this approach impractical.

### Separation of Variables

To solve (2.1–2.3) by separation of variables, the potential is represented as the product of three functions, each depending on only one coordinate [61, 46]. In Cartesian coordinates, for example, substituting the assumed form

$$\psi(x, y, z) = X(x)Y(y)Z(z),$$

reduces (2.1) to three independent ordinary differential equations for  $X(x)$ ,  $Y(y)$  and  $Z(z)$ . The special nature of Laplace's equation ensures that the product representation leads to three separate equations for eleven different coordinate systems [61]. The ordinary differential equation solutions form a parameterized family of  $X(x)Y(y)Z(z)$  products that satisfy (2.1) but not necessarily the boundary conditions (2.2–2.3).

The success of the method hinges on the ability to adjust linear combinations of the product solutions to match the boundary conditions. By construction, each component of the solution corresponding to a particular coordinate can be adjusted independently of the others. In order to exploit this property, the problem's boundaries must align with surfaces that correspond to one coordinate being held constant. Thus separation of variables is difficult to apply when the problem does not align neatly with a separable coordinate system.

The class of separation-of-variables methods known as fast Poisson solvers, however, can treat irregular boundaries [60, 17]. Unfortunately, these methods adjust the solution to match the boundary conditions using the capacitance matrix for the discretized boundary. Calculating the boundary element capacitor information is equivalent to calculating the conductor capacitances. Thus the fast Poisson solver approach is not practical.

### Energy Minimization

The total electrostatic energy stored by the potential  $\psi$ , a solution to (2.1–2.3), is [61]

$$W(\psi) = \frac{\epsilon_0}{2} \iiint_V \|\nabla \psi(x')\|_2^2 dV'. \tag{2.5}$$

Dirichlet's principle states that the potential  $\psi$  that satisfies the boundary conditions (2.2–2.3) and minimizes  $W$  is the solution to (2.1–2.3). The exterior Dirichlet problem can therefore be formulated as a minimization of  $W(\psi)$  over the set of admissible potentials [46].

### Dipole Layer Method

The separation of variables technique leads to harmonic series expansions for the potential. In the case of the multipole expansion of Section 3.2, the series amounts to a combination of the fields due to progressively larger constellations of charges, beginning with a single charge (a monopole) and two charges together (a dipole). Closely related to this approach is superposing the potentials due to a single charge at the origin and dipoles whose strength and position are allowed to vary [82, 63, 69, 51]. Like the series solution, each term in the superposition is harmonic so the problem reduces to adjusting the strengths and positions to match the boundary conditions (2.2–2.3).

If a dipole layer density  $\mu(x)$  replaces the conductor surfaces, then the potential in the free space region  $V$  is given by the superposition integral (see also Appendix A)

$$\psi(x) = \iint_{S_c} \mu(x') \frac{\cos(x - x', n')}{4\pi\epsilon_0 \|x - x'\|_2^2} dS'_c. \quad (2.6)$$

Here  $n'$  is the outward normal to the surface  $S_c$  at  $x'$  and  $\cos(\alpha, \beta)$  is the acute angle between  $\alpha$  and  $\beta$ . Appendix A shows that the potential on the outside surface of the conductors is

$$\psi(x) = \frac{1}{2\epsilon_0} \mu(x) + \iint_{S_c} \mu(x') \frac{\cos(x - x', n')}{4\pi\epsilon_0 \|x - x'\|_2^2} dS'_c, \quad x \in S_c.$$

In particular, since the potential on the conductors is given by  $f(x)$ ,

$$f(x) = \frac{1}{2\epsilon_0} \mu(x) + \iint_{S_c} \mu(x') \frac{\cos(x - x', n')}{4\pi\epsilon_0 \|x - x'\|_2^2} dS'_c, \quad x \in S_c. \quad (2.7)$$

It can be shown that this Fredholm equation of the second kind has no unique solution  $\mu$  for certain specified boundary potentials  $f$  [82, 63, 69]. As a consequence of the nature of the Laplacian operator and the boundary condition at infinity (2.3), the solution to (2.1–2.3) can decrease as slowly as  $1/\|x\|_2$  as  $x$  goes to infinity, but no slower [126]. The potential due to a dipole layer, however, always falls off as  $1/\|x\|_2^2$ , as (2.6) indicates. To model a possible  $1/\|x\|_2$  dependence, it is sufficient to use

$$\psi(x) = \iint_{S_c} \mu(x') \frac{\cos(x - x', n')}{4\pi\epsilon_0 \|x - x'\|_2^2} dS'_c - \frac{1}{2\epsilon_0 \|x\|_2} \iint_{S_c} \mu(x') dS'_c$$

rather than (2.6). The added term corresponds to a charge at the origin. Proceeding as in the previous paragraph gives

$$f(x) = \frac{1}{2\epsilon_0} \mu(x) + \iint_{S_c} \mu(x') \frac{\cos(x - x', n')}{4\pi\epsilon_0 \|x - x'\|_2^2} dS'_c - \frac{1}{2\epsilon_0 \|x\|_2} \iint_{S_c} \mu(x') dS'_c, \quad x \in S_c,$$

a Fredholm equation of the second kind for the unknown dipole layer density  $\mu$  in terms of the specified conductor potentials  $f$ . Unlike (2.7), it can be solved for  $\mu$  given any  $f$  [82, 63, 69, 51]. After it is solved for  $\mu$  the potential in  $V$  can be calculated using (2.6).

### 2.1.2 Capacitance Calculation with Charge Layers

Conductor surfaces may be modeled using surface-charge layers, an approach that is sometimes called the equivalent-charge formulation. In contrast to the techniques of the previous section, the conductor surface-charge density is solved for directly without the use of Gauss's Law. The total charge on each conductor is then calculated by integration. Two approaches are possible, a more general first-kind integral-equation formulation and a second-kind integral formulation, which is only useful for single conductor problems.

#### First-Kind Integral-Equation Formulation

If the conductor surfaces are replaced by a surface-charge layer  $\sigma(x)$  then the potential is given by the superposition integral [126]

$$\psi(x) = \iint_{S_c} \sigma(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|_2} dS'_c \quad (2.8)$$

Appendix A shows that the potential is continuous across the charge layer so that the potential on the surface is

$$\psi(x) = \iint_{S_c} \sigma(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|_2} dS'_c, \quad x \in S_c.$$

Since the potential on the conductors is specified as  $f(x)$  in (2.2),

$$f(x) = \iint_{S_c} \sigma(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|_2} dS'_c, \quad x \in S_c. \quad (2.9)$$

This is a Fredholm integral equation of the first kind for the unknown surface-charge density  $\sigma$  in terms of the known conductor potentials given by  $f$ . Since it relates  $f$  directly to  $\sigma$ , the calculation of the potential throughout  $V$  is avoided.

#### Second-Kind Integral-Equation Formulation

An integral equation of a different form can be derived from the same charge layer assumption [126, 63]. The potential in  $V$  due to the charge layer is given by (2.8) so that the normal derivative at a point  $x$  on the surface is

$$\frac{\partial\psi(x)}{\partial n} = \lim_{s \rightarrow x} \iint_{S_c} \sigma(x') \frac{\partial}{\partial n} \frac{1}{4\pi\epsilon_0 \|s - x'\|_2} dS'_c, \quad (2.10)$$

where  $n$  is the surface outward normal at the point  $x$ ,  $s$  is a point outside of the closed charge layer surface<sup>2</sup> and  $\psi$  is the potential outside of the surface. Applying Gauss's Law to the charge layer gives

$$\sigma(x) = \epsilon_0 \left[ \frac{\partial\psi_{in}(x)}{\partial n} - \frac{\partial\psi(x)}{\partial n} \right].$$

<sup>2</sup>The procedure fails for infinitesimally thin conductors but (2.9) still holds.

Here  $\psi_{in}(x)$  is the potential inside the charge layer shell. Since the charge layer is to model a conductor,  $\psi_{in}$  must be a constant so

$$\sigma(x) = -\epsilon_0 \frac{\partial \psi(x)}{\partial n}. \quad (2.11)$$

With this substitution, Appendix A shows that (2.10) becomes

$$\frac{\partial \psi(x)}{\partial n} = -\frac{1}{\epsilon_0} \sigma(x) = -\frac{1}{2\epsilon_0} \sigma(x) - \iint_{S_c} \sigma(x') \frac{\cos(x - x', n)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS'_c,$$

so that

$$\frac{1}{2\epsilon_0} \sigma(x) = \iint_{S_c} \sigma(x') \frac{\cos(x - x', n)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS'_c. \quad (2.12)$$

This is a homogeneous Fredholm integral equation of the second kind. Any charge layer density shell  $\sigma$  used to model a conductor must satisfy this equation. Since solving (2.12) is an eigenfunction problem,  $\sigma$  is only determined up to a multiplicative constant, provided  $1/2\epsilon_0$  is a simple eigenvalue. For more than one conductor,  $1/2\epsilon_0$  is a repeated eigenvalue, corresponding to many linearly independent  $\sigma$ 's. The space spanned by these charge densities covers all the solutions to the Dirichlet problem for any choice of conductor potentials. The specific charge densities corresponding to one conductor at unit potential, the rest at zero, are difficult to extract from this information.

## 2.2 Numerical Treatment of the Capacitance Problem

Exact, analytic solutions of the capacitance problem using the above formulations are only possible for symmetric geometries. For general problems, approximate solutions must be obtained using some discretization of the formulas. Solution of the algebraic equations resulting from the discretization gives an approximate solution to the capacitance problem.

### 2.2.1 The Method of Weighted Residuals

For concreteness, this method is described as it applies to the first-kind integral equation (2.9),

$$f(x) = \iint_{S_c} \sigma(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|_2} dS'_c.$$

The method of weighted residuals [31, 27, 43] approximates the unknown  $\sigma(x)$  as a linear combination of a set of  $n$  linearly independent expansion functions  $w_i(x)$  with weights  $a_i$ ,

$$\sigma(x) \approx \sum_{i=1}^n a_i w_i(x). \quad (2.13)$$

Substituting the approximation for the charge density into (2.9) gives the residual

$$R(x) \triangleq f(x) - \iint_{S_c} \sum_{i=1}^n a_i w_i(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|_2} dS'_c, \quad (2.14)$$



which is zero when (2.13) holds exactly. For a well-conditioned problem,  $R$  is a good measure of the error in the charge density approximation (2.13), ensuring that minimizing  $R$  leads to similar reductions in the error. The minimization is carried out using a set of  $n$  test functions  $t_i$  whose inner products with the residual are set to zero. Using an inner product defined on the solution space, in this case

$$\iint_{S_c} a(x')b(x')dS'_c,$$

gives  $n$  equations,

$$\iint_{S_c} t_i(x')R(x')dS'_c = 0, \quad i = 1, 2, \dots, n, \quad (2.15)$$

which form an  $n \times n$  linear system for the unknown weights  $a_i$ . Solving the system gives the approximation (2.13). The method of weighted residuals is also called solution by error distribution principles [27], solution by projective methods [123] and the method of moments [57, 69].

Specific choices for the expansion and test functions lead to various special cases of the weighted-residuals approach. The methods are presented in ascending order of numerical complexity.

### Collocation

The residual  $R$  is forced to be zero at  $n$  points  $x_i$  in the solution region,

$$R(x_i) = 0, \quad i = 1, 2, \dots, n.$$

The corresponding test functions are

$$t_i(x) = \delta(x - x_i),$$

where  $x_i$  is the  $i$ -th collocation point. In the case of the capacitance problem the collocation points are distributed over the conductor surfaces. Collocation is also called point collocation or point matching.

### Subdomain Method

The solution domain is divided into  $n$  subdomains,  $S_i$ ,  $i = 1, 2, \dots, n$ , and the integral of the residual over each subdomain is forced to be zero,

$$\iint_{S_i} R(x)dS_i = 0, \quad i = 1, 2, \dots, n.$$

The corresponding  $i$ -th test function is one inside the  $i$ -th subdomain and zero elsewhere,

$$t_i(x) = \begin{cases} 1 & x \in S_i; \\ 0 & \text{otherwise.} \end{cases}$$

Other names for the subdomain method include subdomain collocation and the partition method.

## Galerkin Method

The test functions are identical to the expansion functions,

$$t_i(x) = w_i(x).$$

The inner product equation (2.15) becomes

$$\iint_{S_c} w_i(x) R(x) dS_c = 0, \quad i = 1, 2, \dots, n,$$

which forces the residual to be orthogonal to the space spanned by the expansion functions. Substituting the definition of the residual (2.14) gives

$$\iint_{S_c} w_i(x) \left[ f(x) - \sum_{j=1}^n a_j \iint_{S_c} \frac{w_j(x')}{4\pi\epsilon_0 \|x - x'\|_2} dS'_c \right] dS_c = 0, \quad i = 1, 2, \dots, n.$$

The integral multiplying the unknown  $a_j$  in the  $i$ -th equation is the entry  $P_{ij}$  in the matrix resulting from the Galerkin discretization,

$$\iint_{S_c} w_i(x) f(x) dS'_c = \sum_{j=1}^n a_j P_{ij}, \quad i = 1, 2, \dots, n,$$

with

$$P_{ij} \triangleq \iint_{S_c} w_i(x) \iint_{S_c} \frac{w_j(x')}{4\pi\epsilon_0 \|x - x'\|_2} dS'_c dS_c \quad (2.16)$$

Since also

$$P_{ij} = \iint_{S_c} w_j(x') \iint_{S_c} \frac{w_i(x)}{4\pi\epsilon_0 \|x - x'\|_2} dS_c dS'_c \quad (2.17)$$

$$= P_{ji}, \quad (2.18)$$

the matrix is symmetric. The sequence (2.16-2.18) shows that the inner product of  $w_i$  with the linear operator  $\iint_{S_c} (\cdot) \frac{1}{\|x-x'\|_2} dS'_c$  applied to  $w_j$  is equal to the same quantity with the roles of  $w_i$  and  $w_j$  reversed. This property implies that the linear operator  $\iint_{S_c} (\cdot) \frac{1}{\|x-x'\|_2} dS'_c$  is self-adjoint. In general, any self-adjoint linear operator gives rise to a symmetric Galerkin discretization for any set of expansion functions [31].

## Least-Squares Method

The integral of the residual squared over the solution region is minimized with respect to the unknown expansion weights. In the case of (2.14),

$$\frac{\partial}{\partial a_i} \iint_{S_c} R^2(x) dS_c = 0, \quad i = 1, 2, \dots, n,$$

or

$$\iint_{S_c} \frac{\partial R}{\partial a_i} R dS_c = 0, \quad i = 1, 2, \dots, n. \quad (2.19)$$

Thus the  $i$ -th effective test function in the least-squares approach is the sensitivity of the residual to  $a_i$ . The  $i$ -th inner product equation (2.19) weights the residual more heavily where it has higher sensitivity to the  $i$ -th expansion weight.

### 2.2.2 Stationary-Functional and Finite-Element Methods

The variational formulation of the potential problem minimizes the energy expression (2.5),

$$W(\psi) = \frac{\epsilon_0}{2} \iiint_V \|\nabla\psi(x')\|_2^2 dV', \quad (2.20)$$

over the space of potentials  $\psi$  that satisfy the boundary conditions (2.3). Substituting an approximation

$$\psi(x) \approx \sum_{i=1}^n a_i w_i(x)$$

which satisfies the boundary and differentiability conditions into (2.20) and setting its  $n$  derivatives with respect to the  $a_i$ 's to zero gives the linear system for the expansion weights,

$$\frac{\epsilon_0}{2} \frac{\partial}{\partial a_i} \iiint_V \left\| \sum_{j=1}^n a_j \nabla w_j(x') \right\|_2^2 dV' = 0, \quad i = 1, 2, \dots, n.$$

This procedure is called the Ritz or Rayleigh-Ritz method [31, 27, 43].

The finite-element method [7] began as a Ritz method with expansion functions  $w_i(x)$  taking non-zero values only within finite domains in the solution region. Inside each element the potential is approximated by polynomial functions of the element's vertices. Weighted-residuals methods which use these finite-element expansion functions are also often called finite-element methods, even when the formulation is not of the variational type.

The Ritz method is closely related to the Galerkin method. In general the potential approximation obtained from the variational formulation (2.5) is identical to a Galerkin method applied to the differential problem (2.1-2.3) [43]. Because of this equivalence, the Ritz method is sometimes called the Ritz-Galerkin or Rayleigh-Ritz-Galerkin method, especially when finite-element expansion functions are used.

### 2.2.3 Finite-Difference Methods

The method of weighted residuals can also be applied to the differential equation (2.1-2.3) as long as the approximation for the potential,

$$\psi(x) \approx \sum_{i=1}^n a_i w_i(x),$$

satisfies the boundary conditions and is twice differentiable. In contrast, finite difference methods start with a discrete approximation to the linear operator based on a regular grid covering the solution domain, here the region  $V$  between the conductors [31, 27]. Derivatives are replaced by divided differences relating approximations to the potential at nearby grid points. For a regular cartesian grid, the potential at each point,  $\psi(x_i, y_i, z_i)$ , can be used along with its six neighboring points to write six approximate directional derivatives of the potential,

$$\frac{\partial\psi}{\partial x_{i+}} \triangleq \frac{\psi(x_i + h, y_i, z_i) - \psi(x_i, y_i, z_i)}{h}; \quad \frac{\partial\psi}{\partial x_{i-}} \triangleq \frac{\psi(x_i - h, y_i, z_i) - \psi(x_i, y_i, z_i)}{h};$$

$$\frac{\partial \psi}{\partial y_{i+}} \triangleq \frac{\psi(x_i, y_i + h, z_i) - \psi(x_i, y_i, z_i)}{h}, \quad \frac{\partial \psi}{\partial y_{i-}} \triangleq \frac{\psi(x_i, y_i - h, z_i) - \psi(x_i, y_i, z_i)}{h},$$

$$\frac{\partial \psi}{\partial z_{i+}} \triangleq \frac{\psi(x_i, y_i, z_i + h) - \psi(x_i, y_i, z_i)}{h}, \quad \frac{\partial \psi}{\partial z_{i-}} \triangleq \frac{\psi(x_i, y_i, z_i - h) - \psi(x_i, y_i, z_i)}{h}.$$

These quantities are combined in approximate Gauss's Law equations for small cubic volumes enclosing the  $n$  grid points, giving the linear system

$$\frac{\partial \psi}{\partial x_{i+}} + \frac{\partial \psi}{\partial x_{i-}} + \frac{\partial \psi}{\partial y_{i+}} + \frac{\partial \psi}{\partial y_{i-}} + \frac{\partial \psi}{\partial z_{i+}} + \frac{\partial \psi}{\partial z_{i-}} = 0, \quad i = 1, 2, \dots, n.$$

In this way the finite difference method enforces the approximation to (2.1–2.3) locally. Thinking of the approximate  $\nabla^2 \psi$  as a residual, the local finite difference equations can be viewed as the result of inner products with test functions  $t_i$  that are non-zero only near the  $i$ -th grid point. A more careful analysis shows that the finite difference approximation is equivalent to a Galerkin method [43].

## 2.3 Previous Work

Existing capacitance calculation methods vary from numerically efficient techniques for special problems to computationally intensive general algorithms.

### 2.3.1 Methods for Special Geometries

Highly specialized approaches include those leading to separation of variables solutions to (2.1–2.3) [73, 18, 16], perturbation and imaging techniques where a known solution is adjusted to solve a similar problem [40, 132, 77, 22, 23] and methods for special periodic structures [20, 136, 101]. Less restrictive but still specialized are two-dimensional approaches, motivated largely by transmission line analysis [138, 122, 26, 103, 35, 44, 51, 21, 114, 106], techniques for three-dimensional periodic structures [143, 142, 118] and on-chip capacitance extraction methods that piece together precomputed solutions [79] or assume planar geometries [100, 6, 8, 95, 134].

### 2.3.2 Methods for General Three-Dimensional Geometries

In cases with no symmetry, the general three-dimensional problem is usually solved by finite-difference [144, 117, 55, 145], finite-element [5, 29] or weighted-residuals [41, 9, 109, 63, 105, 130, 115, 96, 95, 71] methods. Finite-difference and finite-element schemes applied to (2.1–2.3) require grid point meshes that are difficult to construct for irregular conductor geometries. Also, both methods discretize the entire region  $V$  between the conductors and must terminate the discretization so as to approximately match the boundary conditions at infinity. In some cases this shortcoming can be mitigated by special techniques [145]. The differential equation (2.1–2.3) can also be solved by collocation, with collocation points distributed throughout  $V$

[71]. All these indirect methods require a separate step to calculate the conductor charge from the potential with Gauss's Law.

In general the simplest techniques are weighted residuals methods applied to first-kind integral-equation formulations like (2.9), which solve for the charge density directly rather than first finding the potential and then applying Gauss's Law. The three-dimensional problem is reduced to finding the two-dimensional surface-charge density. Expansion functions are usually of the finite-element type [41, 9, 109, 105, 130, 115] or sometimes tailored to particular geometries [95]. Collocation [41, 105] and the intermediate complexity subdomain [42] and Galerkin [9, 109, 115, 96, 95] techniques are both used, more often than least-squares methods.

The algorithm studied in this thesis uses a weighted residuals discretization of the integral-equation formulation (2.9), leading to a linear system whose solution gives the charge density directly. The use of collocation, following [105] as described in Section 3.1, simplifies the discretization, allowing the treatment of arbitrarily shaped geometries. The major departure of this work from previous approaches is in the use of an efficient approximate solution method for the discretized problem [88, 83, 91, 85]. The solution technique, described in Section 3.2, is based on [107, 52, 54] and is also used in the two-dimensional, dipole-layer method of [51]. While earlier approaches lead to efficient algorithms by restricting the problem geometry, the acceleration technique used here exploits the basic physics of the capacitance problem and applies to any three-dimensional geometry, including those with multiple dielectrics, as Section 3.3 demonstrates.



---



---

# Multipole Accelerated Capacitance Calculation

The capacitance-calculation algorithm developed in this thesis starts with a weighted-residuals discretization of the first-kind integral-equation formulation (2.9). The method uses collocation on conductor-surface panels as described in Section 3.1, giving a linear system for the panel charges. The major contribution of this thesis is the use of a multipole-accelerated, preconditioned iterative method, described in Section 3.2, to solve the resulting linear system. For an  $m$ -conductor problem discretized with  $n$  panels, existing Gaussian-elimination methods require nearly  $O(n^3)$ <sup>1</sup> time and nearly  $O(n^2)$  storage, while the new method uses nearly  $O(mn)$  time and  $O(n)$  storage. Section 3.3 shows how the same savings are possible for problems with multiple isotropic dielectric regions.

## 3.1 First-Kind Integral-Equation Discretization

A weighted-residuals method applied to the integral equation (2.9) is a simple and efficient way to solve the capacitance problem [105]. A finite-element expansion of the conductor surface-charge densities followed by collocation leads directly to a linear system for the charge densities.

### 3.1.1 Weighted-Residuals Discretization

The unknown conductor surface-charge density is approximated by a mosaic of two-dimensional finite elements. Each expansion function is non-zero only in a small triangular or quadrilateral region. These regions are called panels or tiles and completely cover the conductor surfaces. Similar techniques can be applied to the more general multiple-dielectric formulation (3.36) and (3.39) described in Section 3.3.

The  $n$  expansion functions are taken as unknown constants  $a_j$  on their corresponding panels

---

<sup>1</sup>When  $f(x) \neq 0$ ,  $g(x)$  is  $O(f(x))$  if there is a constant  $c$  such that  $\lim_{x \rightarrow \infty} g(x)/f(x) < c$ .

and zero elsewhere. Thus the approximate charge density has the form

$$\sigma(x) \approx \sum_{j=1}^n a_j w_j(x), \quad (3.1)$$

where

$$w_j(x) \triangleq \begin{cases} 1 & x \in \text{panel } j; \\ 0 & \text{otherwise.} \end{cases}$$

Substitution gives the residual (2.14),

$$R(x) = f(x) - \iint_{S_c} \sum_{j=1}^n \frac{a_j w_j(x')}{4\pi\epsilon_0 \|x - x'\|} dS'_c,$$

where  $f$  specifies the conductor potentials. Collocation produces  $n$  equations by requiring the potential at the panel centroid points  $x_i$  to be the specified value, effectively setting the integral of  $\delta(x' - x_i)R(x)$  over  $S_c$  to zero for each  $x_i$ . Other collocation point choices are possible, for example panel corners. The equation for the  $i$ -th collocation point is of the form

$$0 = f(x_i) - \sum_{j=1}^n \iint_{S_c} \frac{a_j w_j(x')}{4\pi\epsilon_0 \|x_i - x'\|} dS'_c.$$

Because  $a_j$  is constant and  $w_j$  is non-zero only on the  $j$ -th panel,

$$\sum_{j=1}^n a_j \iint_{\text{panel } j} \frac{1}{4\pi\epsilon_0 \|x_i - x'\|} dS'_c = f(x_i), \quad i = 1, 2, \dots, n.$$

The constant  $a_j$  has the physical interpretation of a constant charge density on panel  $j$ ,

$$a_j = \frac{q_j}{A_j},$$

where  $q_j$  and  $A_j$  are the total charge and area of panel  $j$ , respectively. This substitution gives an  $n \times n$  linear system for the panel charges in terms of the conductor potentials,

$$\sum_{j=1}^n q_j \left( \frac{1}{A_j} \iint_{\text{panel } j} \frac{1}{4\pi\epsilon_0 \|x_i - x'\|} dS'_c \right) = f(x_i), \quad i = 1, 2, \dots, n, \quad (3.2)$$

or, in matrix form,

$$Pq = f. \quad (3.3)$$

Here  $q$  is the vector of panel charges and

$$\begin{aligned} P_{ij} &\triangleq \frac{1}{A_j} \iint_{\text{panel } j} \frac{1}{4\pi\epsilon_0 \|x_i - x'\|} dS'_c; \\ f_i &\triangleq f(x_i). \end{aligned} \quad (3.4)$$

The  $P_{ij}$ 's are called potential coefficients and can be calculated in closed form for all panel orientations [94]. The corresponding quantities in the Galerkin case, given by (2.16), have closed



form expressions only in special cases [109], although the resulting linear system generally leads to more accurate capacitances. Using panel-centroid collocation points makes the  $P_{ij}$ 's first-order approximations to the Galerkin potential coefficients, adding some accuracy compared to using other collocation point positions. In general the accuracy of the weighted-residuals method can be enhanced using subdomain, Galerkin, or possibly hybrid, sets of test functions combined with variable order finite element interpolating polynomials. The acceleration technique described in the next section also applies to the discretized system obtained using any of these approaches.

### 3.1.2 Discretization Build and Solve Costs

Solving (3.3) gives the piecewise-constant approximation to the charge density. Summing the charge vector entries corresponding to panels on conductor  $i$ , for the potential problem with conductor  $j$  at unit potential, gives the capacitance matrix entry  $C_{ij}$ . For an  $m$  conductor problem, (3.3) must be solved  $m$  times with  $m$  different right-hand-side vectors, each solution yielding a column of  $C$ .

The dominant costs of this algorithm are the computation of  $P$  and the solution of the  $m$   $Pq = f$  systems. In the standard approach, every entry of  $P$  is formed using time and memory proportional to  $n^2$  to evaluate and store the  $n^2$  inner product integrals. Factoring  $P$  by variants of Gaussian elimination takes time proportional to  $n^3$  since  $P$  is dense. With  $P$  factored, the  $m$   $Pq = f$  solutions cost additional time proportional to  $mn^2$ . The acceleration technique described in the next section reduces the cost of all these calculations to roughly  $O(mn)$  time and  $O(n)$  memory.

## 3.2 Accelerating the Discretized System Solution

The main contribution of this thesis is the acceleration of the capacitance matrix calculation for an arbitrary three-dimensional conductor system. The accelerated method is based on the standard weighted-residuals discretization of the integral-equation formulation (2.9) described in Section 3.1. The problem is reduced to solving  $m$   $n \times n$  linear systems of the form

$$Pq = f,$$

by expanding the surface-charge density on the  $m$  conductors in  $n$  functions. Although similar techniques apply for subdomain and Galerkin methods, collocation is assumed here.

The acceleration technique reduces the combined  $O(n^3)$  solution time and  $O(n^2)$  time required to calculate  $P$  to time nearly  $O(mn)$ . Using an iterative method reduces the solution time to roughly  $O(mn^2)$  as described in Section 3.2.1. The iterative method time is then reduced to nearly  $O(mn)$  using the fast multipole algorithm as discussed in Section 3.2.2.

The idea of solving potential problems using an iterative method coupled with the multipole algorithm is due to Rokhlin [107], although the need for an approximate technique in the context of capacitance calculations was first recognized by Ruehli, Brennan and Young [110]. The three-dimensional fast multipole algorithm is due to Greengard [52, 54].

### 3.2.1 Solving the Discretized System Iteratively

Many iterative methods for the linear system  $Pq = f$  follow the same basic procedure.

**Algorithm 3.1 (Generic Iterative Method)** *To solve the  $n \times n$  linear system  $Pq = f$  with a sequence of convergent iterates  $q^i$ ,*

1. *Guess at a solution  $q^0$ . Set  $i = 0$ .*
2. *Calculate  $Pq^i$ .*
3. *If the residual  $f - Pq^i$  is small enough, return  $q^i$  as the solution. Otherwise adjust  $q^i$  to obtain  $q^{i+1}$ , increment  $i$  and return to step 2.*

The time required for one iteration is dominated by the  $O(n^2)$  operations required to perform the matrix-vector product in step 2. If it can be shown that the number of iterations required for a given accuracy is bounded by a constant independent of  $n$ , then the entire solution requires  $O(n^2)$  operations. The matrix problems  $Pq = f$  arising from the collocation based solution of the capacitance-problem integral equation (see Section 3.1) appear to have this property, as the results in Section 4.4 indicate. In contrast, solving  $Pq = f$  using Gaussian elimination procedures requires  $O(n^3)$  operations.

The matrix-vector product  $Pq^i$  is the potential due to the panel charges  $q^i$  at all the collocation points. The fast multipole algorithm can be used to quickly calculate such potentials, making it important that the iterative method have this form.

### Krylov Subspace Iterative Methods

An important class of iterative methods that have the form of Algorithm 3.1 are called Krylov subspace or polynomial methods. The most effective of these methods are inspired by the conjugate-gradient algorithm (CG), an iterative method for symmetric, definite  $P$  [58]. The  $i$ -th CG iteration minimizes the error in the  $i$ -th iterate  $q^i$  obtained from a linear combination of the first  $i$  vectors of the associated Krylov space<sup>2</sup>. Because of the form of the Krylov vectors, the linear combinations equal to  $q^i$  and its error  $q - q^i$  may be written as polynomials in  $P$  multiplying the initial residual vector  $f - Pq^0$  (see also Section 3.2.4). As the iteration progresses, the error polynomial approaches a scaled version of  $P$ 's characteristic polynomial. Thus CG

<sup>2</sup>If the initial guess is the zero vector, the Krylov space vectors are  $f, Pf, P^2f, \dots, P^{n-1}f$ .

convergence is related to  $P$ 's eigenvalues, with clusters of eigenvalues generally implying faster convergence. Convergence in at most  $n$  iterations is guaranteed since the error minimization is eventually done throughout  $R^n$  once the last Krylov vector is included. The implementation of the algorithm requires only a three-term linear recurrence and the matrix-vector product.

The matrix  $P$  obtained by collocation is not symmetric, nor is it guaranteed to be definite. The use of the multipole algorithm to approximate  $Pq^i$  products amounts to multiplying  $q^i$  by a generally unsymmetric, indefinite approximation to  $P$ , even when using a Galerkin method. A variant of CG that works for unsymmetric, indefinite  $P$  is required. Unsymmetric Krylov space methods fall into two broad classes: those that retain the efficient three-term recurrence of CG and those that retain the minimization and finite termination properties of CG. The lack of symmetry makes it difficult to obtain both properties simultaneously. The three-term recurrence algorithms include the bi-conjugate gradient (Bi-CG) method and the conjugate gradient squared (CGS) method [125]. Bi-CG is more difficult to implement since it requires multiplication by  $P^T$ , an operation that is not so readily interpreted as a potential calculation. The sparse least squares (LSQR) [98], generalized conjugate residual (GCR) and generalized minimum residual (GMRES) [112] algorithms all minimize the Euclidean norm of the residual,  $\|f - Pq^i\|_2$ , over the current Krylov subspace on each iteration, but can require significant overhead since they lack the three-term recurrence. LSQR, like Bi-CG, requires  $P^Tq$  products but is thought to be more stable when  $P$  is poorly conditioned.

The overhead in LSQR, GCR and GMRES involves inner products that maintain the orthogonality of the current adjustment to  $q^i$  to those adjustments made on previous iterations. Computing the  $(i + 1)$ -th iterate requires at least  $i$  such inner products at a cost of  $n$  multiply-adds each. For sparse system matrices, a few inner products cost as much as a matrix-vector product, making methods that maintain the three-term recurrence attractive. For the capacitance-calculation method developed here, the cost of the matrix-vector products, although  $O(n)$  when approximated with the multipole algorithm, is still much more than the inner-product overhead, since convergence is typically rapid. Thus using a three-term-recurrence method rather than LSQR, GCR or GMRES sacrifices residual-minimization properties for a negligible savings in execution time. Since LSQR uses  $P^Tq$  products, the algorithms of this thesis use the simpler GMRES, an improved version of GCR. The extra stability of LSQR is unnecessary for the well-conditioned matrices encountered in capacitance problems. In order to obtain well-conditioned matrices for many practical problems, however, it is necessary to use a preconditioner, as the next section describes.

## Preconditioning

Preconditioning is a common technique for accelerating iterative method convergence. In the case of preconditioned CG, the system

$$(B^{-1}PB^{-1})(Bq) = B^{-1}f,$$

rather than  $Pq = f$ , is solved iteratively [50]. Because of the CG method's three-term recurrence, the calculation can be arranged so that the iterates tend toward  $q$  rather than  $Bq$ , and  $B^{-1}PB^{-1}$  and  $B^{-1}f$  need never be formed explicitly. Similar techniques are possible for preconditioned CGS, since its iterates are given by a three-term recurrence as well. However, lack of symmetry in  $P$  forces the use of the preconditioned system  $(B_L^{-1}PB_R^{-1})B_Rq = B_L^{-1}f$ . The choice of  $B_L$  and  $B_R$  is generally difficult since only crude analytical techniques exist for predicting CGS convergence [125].

The implementation of preconditioned GMRES used in this thesis solves

$$\tilde{C}Pq = \tilde{C}f,$$

rather than  $Pq = f$ , following the strategy of CGS preconditioning<sup>3</sup>. Since GMRES lacks the three-term recurrence of CGS, the iterative method must explicitly compute the scaled right-hand-side,  $\tilde{C}f$ . An advantage of GMRES over CGS is that its convergence properties for normal matrices<sup>4</sup> are linked to the eigenvalues of the matrix, much like the CG case [112]. In the non-normal case similar evaluation techniques apply [131]. Thus the effectiveness of the preconditioner matrix  $\tilde{C}$  can be gauged by how it changes the eigen-structure seen by GMRES when it is used to solve the system  $\tilde{C}Pq = \tilde{C}f$ . In general preconditioners that lead to clustered eigenvalues, with  $\tilde{C} = P^{-1}$  the limiting case, are better. For the capacitance algorithm presented here, the use of the multipole algorithm suggests using the part of  $P$  corresponding to nearby panels to form a preconditioner. Section 3.2.4 discusses this strategy further.

### Block Iterative Methods

The capacitance calculation algorithm for an  $m$  conductor problem amounts to solving  $Pq = f_k$ ,  $k = 1, 2, \dots, m$ , where  $f_k$  specifies conductor  $k$  at unit potential, the rest at zero potential. Such a multiple right-hand-side problem can usually be solved more efficiently by a block iterative technique [97, 111] while still preserving compatibility with the multipole algorithm. A block iterative method uses iterates that are blocks of vectors. If  $q_k$  is the solution corresponding to  $f_k$  then the capacitance-problem block system is

$$PQ = F,$$

with

$$Q \triangleq \begin{bmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_m \\ | & | & & | \end{bmatrix}; \quad F \triangleq \begin{bmatrix} | & | & & | \\ f_1 & f_2 & \cdots & f_m \\ | & | & & | \end{bmatrix}.$$

<sup>3</sup>Solving the preconditioned system  $P\tilde{C}x = f$  for  $x$  and then calculating  $q = \tilde{C}x$ , as advocated in [83], is not as desirable since  $\tilde{C}$  can amplify small errors in  $x$  to produce large errors in  $q$ .

<sup>4</sup>A real matrix  $P$  is normal if  $PP^T = P^T P$  or, equivalently, if it has a full set of orthonormal eigenvectors [127].

**Algorithm 3.2 (Generic Block Iterative Method)** *To solve the  $n \times n$  block linear system  $PQ = F$  with a sequence of convergent iterates  $Q^i$ ,*

1. *Guess at a solution  $Q^0$ . Set  $i = 0$ .*
2. *Calculate  $PQ^i$ .*
3. *If the residual  $F - PQ^i$  is small enough, return  $Q^i$  as the solution. Otherwise adjust  $Q^i$  to obtain  $Q^{i+1}$ , increment  $i$  and return to step 2.*

In the case of block CG, the error in the block iterate  $Q^i$  is minimized over the current partial block Krylov space<sup>5</sup> in a manner exactly analogous to the single-vector CG method. Similar analogies exist between the scalar and block versions of GMRES. A single block iteration is more costly than  $m$  single-vector iterations, particularly because the scalar recursion weights become  $m \times m$  matrices in the block implementation. These matrices, however, couple the columns of the error  $Q - Q^i$ , often leading to faster convergence overall.

The basic convergence properties of the single-vector iterations carry over, with some enhancement, to the block methods. Thus preconditioners are analyzed and applied as for the single-vector cases. For example, a preconditioned block GMRES method would effectively apply the block GMRES algorithm to

$$\bar{C}PQ = \bar{C}F.$$

Section 5 further examines using a preconditioned block-GMRES iterative method for capacitance calculation.

### 3.2.2 Efficient Matrix-Vector Products with the Multipole Algorithm

The key to accelerating the iterative algorithms of Section 3.2.1 is interpreting the matrix-vector product  $Pq = f$  as a potential calculation. The  $i$ -th entry of the vector  $f$ ,

$$f_i = \sum_{j=1}^n P_{ij}q_j, \quad (3.5)$$

is the potential at the  $i$ -th collocation point due to the panel charges  $q_j$ ,  $j = 1, \dots, n$ . Assuming all the panels have comparable charges and geometries, the largest contributions to  $f_i$  come from panels near the  $i$ -th collocation point, with distant panels contributing less.

The fast multipole algorithm effectively replaces the part of the inner product (3.5) corresponding to distant panels with an approximation [52]. Thus  $P_{ij}$  is no longer required if panel  $j$  is far enough away from collocation point  $i$ . The entry  $P_{ij}$  need not be computed, stored nor

---

<sup>5</sup>If the initial guess is the zero matrix, the block Krylov space matrices are  $F, PF, P^2F, \dots, P^{n-1}F$ .

explicitly multiplied by  $q_j$ . In this way the multipole algorithm reduces both execution time and storage requirements.

The potential approximation used by the multipole algorithm is similar to that used by Hackbusch and Nowak [56]. The multipole algorithm, however, is more efficient since it evaluates the potential approximation in a manner that eliminates redundant calculations (see Section 3.2.2). Perhaps the simplest approximation for the distant panels potentials is to assume they are all zero, as advocated by Dewilde and his collaborators for planar geometries [95, 134, 38]. In contrast, the multipole approximation is always a function of problem geometry and charge distribution, and never completely ignores distant panel potentials.

### The Multipole Expansion

The multipole algorithm approximates the potential  $\psi$  due to a collection of charged panels using a multipole expansion

$$\psi(r_i, \theta_i, \phi_i) \approx \frac{1}{4\pi\epsilon} \sum_{n=0}^l \sum_{m=-n}^n \frac{M_n^m}{r_i^{n+1}} Y_n^m(\theta_i, \phi_i), \quad (3.6)$$

where  $l$  is the expansion order,  $\epsilon$  is the permittivity of the surrounding dielectric and  $r_i$ ,  $\theta_i$ , and  $\phi_i$  are the spherical coordinates of the  $i$ -th panel centroid<sup>6</sup> relative to the center of the smallest sphere enclosing all the distant panels. The  $Y_n^m(\theta_i, \phi_i)$  factors are called surface spherical harmonics [59, 66] and the  $M_n^m$  are complex weights known as multipole coefficients given by

$$M_n^m = \sum_{j=1}^d \frac{q_j}{A_j} \iint_{\text{panel } j} (\rho')^n Y_n^{-m}(\alpha', \beta') dS'_c,$$

where there are  $d$  distant panels with constant charge densities  $q_j/A_j$  as in (3.2). The variables of integration are the spherical coordinates  $(\rho', \alpha', \beta')$  of the differential panel surface area  $dS'_c$  relative to the center of the smallest sphere enclosing all the distant panels.

An approximation to the contribution to the potential at the  $i$ -th collocation point  $x_i$  due to the  $d$  distant panel charges is obtained by evaluating the multipole expansion at  $x_i$ . Many other potential evaluations at collocation points at least as far away from the  $d$  panels as  $x_i$ , can use the same multipole expansion to include the effect of the distant panels. Assuming there are  $d$  collocation points and the multipole expansion costs nothing to compute, this procedure, illustrated in Figure 3-1, costs  $O(d)$  operations. This is an improvement over the  $O(d^2)$  cost of directly evaluating the potential due to  $d$  panel charges at  $d$  collocation points. The error in the approximation of Figure 3-1 is [52],

$$\left| \psi(r_i, \theta_i, \phi_i) - \frac{1}{4\pi\epsilon} \sum_{n=0}^l \sum_{m=-n}^n \frac{M_n^m}{r_i^{n+1}} Y_n^m(\theta_i, \phi_i) \right| \leq K \left( \frac{R}{r_i} \right)^{l+1} \leq K \left( \frac{R}{r} \right)^{l+1}. \quad (3.7)$$

<sup>6</sup>Collocation points on panel centroids are assumed here.

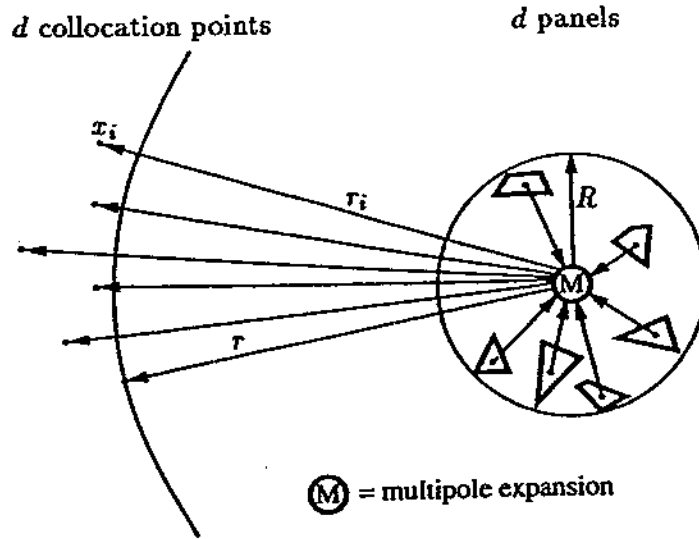


FIGURE 3-1: Evaluation of  $d$  distant panel potentials at  $d$  collocation points with the multipole expansion.

The quantities  $r$  and  $R$  are as in Figure 3-1 and

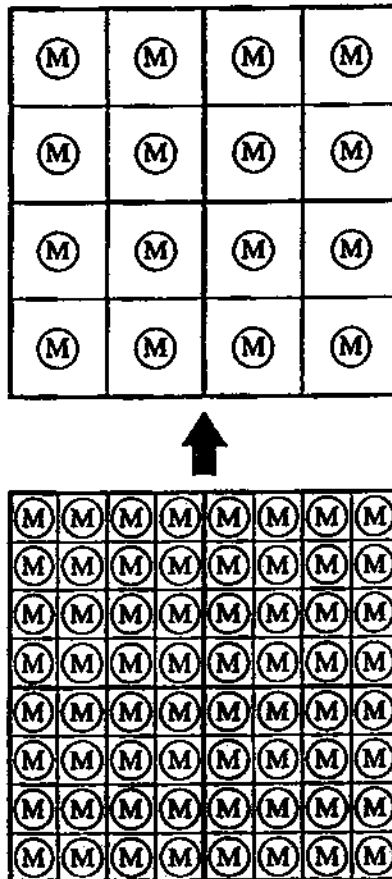
$$K \triangleq \sum_{j=1}^d |q_j| / [4\pi\epsilon(r - R)] \quad (3.8)$$

is independent of the multipole-expansion order,  $l$ . The length  $r$  is the distance to the nearest collocation point from the multipole expansion center and  $R$  is the radius of the sphere enclosing the panel charges.

### The Partitioning of Space

The aggregation of distant panels into multipole expansions which can be used to evaluate potentials at many collocation points is the source of the fast multipole algorithm's efficiency. Maintaining this efficiency for general panel and collocation point distributions is made possible by the remaining parts of the algorithm. The organizational framework for the construction and evaluation of the multipole expansions is provided by a hierarchical partitioning of the problem domain, defined to be the smallest cube containing all the conductors. The cube is recursively divided into eight smaller cubes, giving  $8^D$  cubes after  $D$  partitioning levels. A given partitioning of space has cubes on levels  $0, 1, \dots, D$ , with the cubes decreasing in size as the level number increases. Every cube  $i$  except the level 0 cube has a parent cube, the cube that is partitioned to give cube  $i$ . Any cube other than the level 0 cube is called a child cube of its parent cube.

### The Upward Pass



**FIGURE 3-2: The combining of lower-level cube multipole expansions to form multiple expansions for their parent cubes.**



In the first part of the multipole algorithm, a multipole expansion for every cube at every level is calculated. After this step, the multipole expansions for every possible distant panel evaluation are in place. The calculation procedure is called the upward pass since it starts with the level  $D$  (smallest) cubes and works up the hierarchy from child to parent. Figure 3-2 illustrates how one upward-pass step combines sets of four (eight in three dimensions) child-cube expansions on the finer level to form single parent-cube expansions, giving a coarser scale representation of all the panel charge. A parent multipole expansion contains no new information; it merely summarizes the effects of all the panels in its children.

### Direct Evaluation of Multipole Expansions

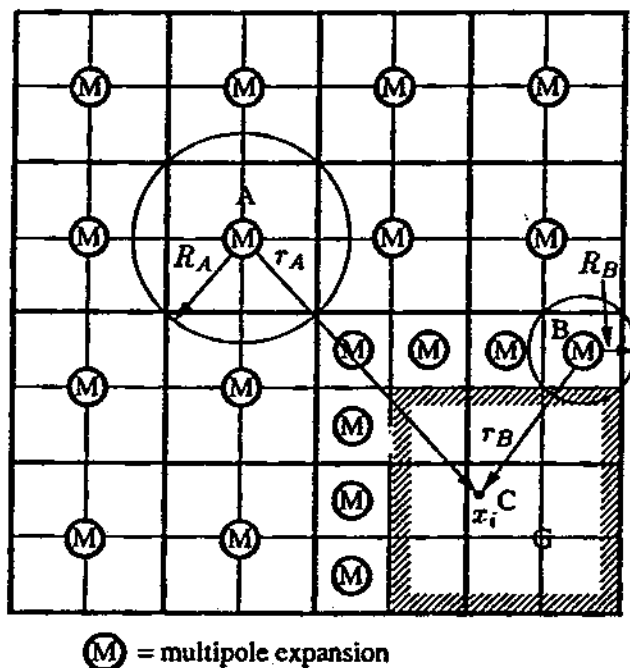


FIGURE 3-3: Evaluation of the potential at  $x_i$ .

The evaluation process uses the partitioning hierarchy as a means of minimizing both the number of multipole evaluations and the error. Figure 3-3 illustrates the procedure in two dimensions for ease of presentation. The figure shows how distant-panel multipole expansions from different sized cubes in the hierarchy contribute to the potential at the collocation point  $x_i$ . In order to minimize the number of multipole evaluations, it is advantageous to use expansions corresponding to larger cubes, such as the larger circled cube A. However, in order to keep the multipole-expansion truncation error roughly the same for all the evaluations, the distant panels closer to  $x_i$  must be included using expansions corresponding to smaller cubes. For example, (3.7) predicts roughly equal error when the cube A and B multipole expansions are evaluated since  $R_A/r_A \approx R_B/r_B$ . In the three-dimensional fast multipole algorithm, the partitioning geometry is used as a guide to selecting multipole expansions such that the ratio  $R/r$  in (3.7)

satisfies

$$R/\tau < 1/\sqrt{3}, \quad (3.9)$$

implying significantly tighter error bounds for each increase in expansion order [52].

The panels in the region with the shaded border cannot be considered distant. Their contribution to the potential at  $x_i$  must be calculated directly from the panel charges using a sum of  $P_j q_j$  products.

### Evaluation of Multipole Expansions with Local Expansions

In the scenario of Figure 3-3, cube C contains the single collocation point  $x_i$ , but usually a given cube like C contains several collocation points. The distant panel contributions to the potentials at each of the collocation points in the given cube are nearly identical, as the points are close together. This suggests the use of a local approximation to the distant multipole potentials, an approximation that is accurate within the small given cube region. The potentials of the many distant panel multipole expansions are condensed into a single local expansion, reducing the number of expansions to evaluate at each collocation point from many to one.

The three-dimensional fast multipole algorithm converts the many multipole expansions into a single local expansion

$$\psi(r_i, \theta_i, \phi_i) \approx \frac{1}{4\pi\epsilon} \sum_{n=0}^l \sum_{m=-n}^n L_n^m Y_n^m(\theta_i, \phi_i) r_i^n, \quad (3.10)$$

where  $l$  is the order of the expansion,  $r_i$ ,  $\theta_i$  and  $\phi_i$  are the spherical coordinates of the  $i$ -th collocation point with respect to the cube's (analogous to cube C) center and the  $L_n^m$  factors are the complex local expansion coefficients. The procedure is illustrated schematically in Figure 3-4. Several distant panel multipole expansions, corresponding to cubes of different sizes, are used to produce a single local expansion which is evaluated at all the evaluation-cube collocation points.

As for the multipole expansion, the error introduced by the local expansion is related to a ratio of distances,

$$\left| \psi(r_i, \theta_i, \phi_i) - \frac{1}{4\pi\epsilon} \sum_{n=0}^l \sum_{m=-n}^n L_n^m Y_n^m(\theta_i, \phi_i) r_i^n \right| \leq K \left( \frac{r_i}{r} \right)^{l+1} \leq K \left( \frac{R}{r} \right)^{l+1}. \quad (3.11)$$

Here  $K = \sum_{j=1}^d |q_j|/[4\pi\epsilon(r-R)]$ , as for the constant in the multipole-expansion error expression (3.8), and  $\epsilon$  is the permittivity of the surrounding medium. The distances  $r$  and  $R$  are as in Figure 3-4. The length  $r$  is the distance to the nearest panel included in the multipole expansions and  $R$  is the radius of the sphere of collocation points. The partitioning and evaluation rules used by the fast multipole algorithm ensure that (3.9) holds [52].

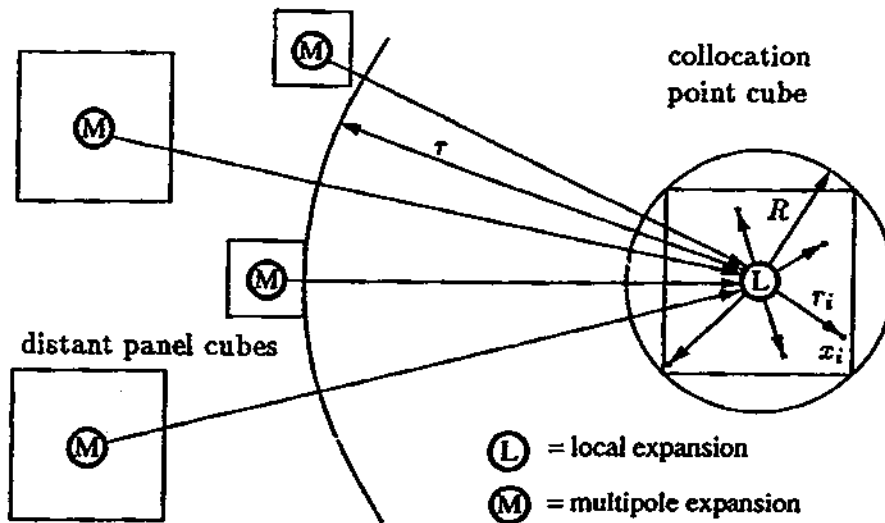


FIGURE 3-4: The conversion of several distant multipole expansions into one local expansion for evaluation at the local collocation points.

### The Downward Pass

The key to the fast multipole algorithm's efficiency lies in the calculation of lowest-level local expansions (as for cube C in the Figure 3-3 example) using a hierarchy of higher level local expansions. As illustrated in Figure 3-5a, each expansion-hierarchy construction step converts coarser, parent-cube expansions into expansions for their children. The resultant downward pass through the cube hierarchy moves from parent to child, following the path of the upward pass in reverse.

The analogy to the upward pass, however, is less than perfect because a child cube's local expansion is not completely determined by its parent cube's expansion. Figure 3-5b illustrates the conversion of the local expansion for cube G to the expansion for one of its children, cube C. A local expansion summarizes the effects of distant panel charges. In the context of the fast multipole algorithm, the local expansion for any cube at any level summarizes the effects of all panel charge outside the cube's neighbor region<sup>7</sup>. In the example of Figure 3-5b, cube G's local expansion summarizes the illustrated higher level multipole expansions representing charge panels outside its shaded neighbor region. The local expansion for G's child cube C must, like cube G's local expansion, represent the effect of all distant charge. Since C is on a finer level, its neighbor region, shaded in Figure 3-5b, is smaller, so C's local expansion must include the effects of not only all the multipole expansions summarized by G's local expansion, but also the lower-level expansions corresponding to the part of G's neighbor region that is not also part of C's neighbor region, those multipole expansions illustrated in the lower part of Figure 3-5b.

<sup>7</sup>Section 4.1.2 gives a precise definition of neighbor cubes for the three-dimensional algorithm.

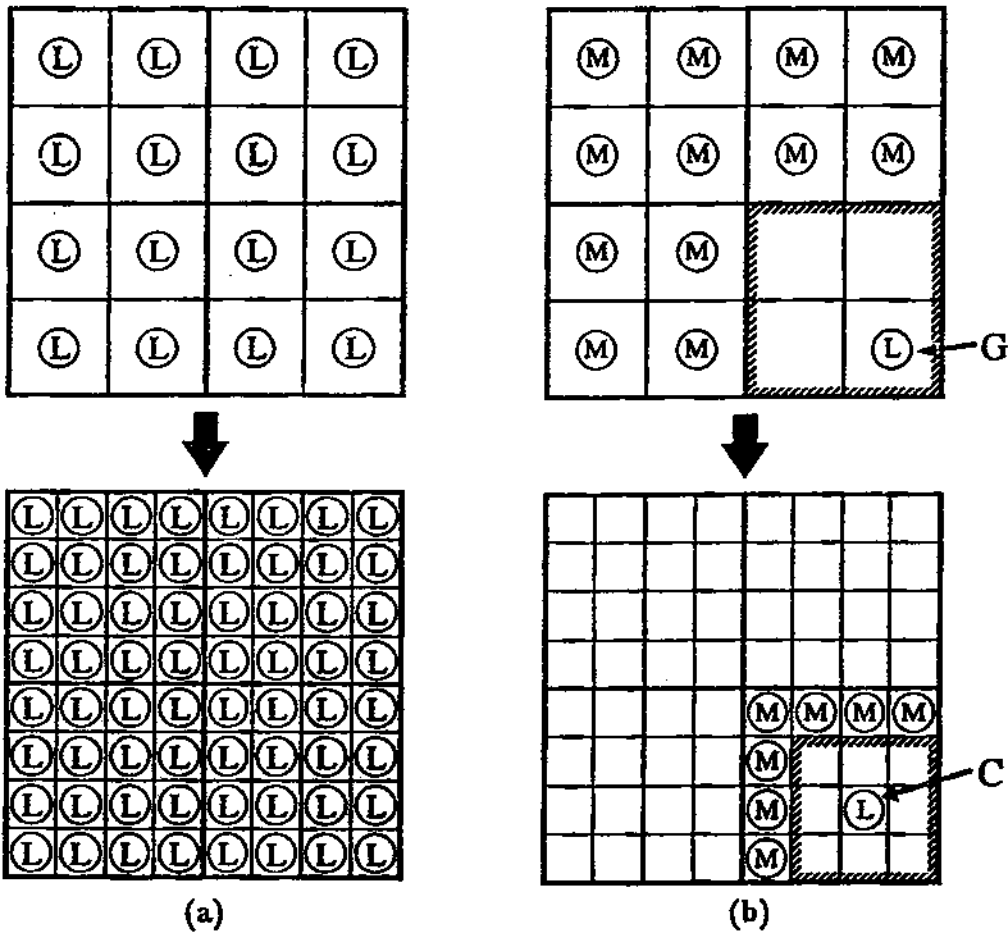


FIGURE 3-5: One step (a) in the construction of the local-expansion hierarchy consists of building many lower-level child cube expansions using the upper-level parent expansions and distant multipole expansions as in (b).

Rather than explicitly converting all these multipole expansions to a local expansion for C, the fast multipole algorithm starts with G's local expansion shifted to C's center to include the larger cube multipole expansions, and adds explicit conversions of only the smaller, medium-distance cubes' multipole expansions. The non-empty cubes in the medium-distance region, those cubes labeled as containing multipole expansions in the lower part of Figure 3-5b, make up C's interaction list or set<sup>8</sup>. The downward pass begins with cubes on level 2, since cubes on levels 1 and 0 have null interaction sets, and stops on the lowest level.

### The Evaluation Pass

The complete three-dimensional multipole algorithm constructs multipole expansions for all the cubes level 2 or lower using the upward pass procedure. The downward pass then converts the multipole expansions to local expansions for the lowest level cubes. The final step, the evaluation pass, consists of evaluating the local expansions at the collocation points and calculating the neighbor-cube contributions directly. The entire procedure requires  $O(n)$  time and storage (see also Section 4.3) and brings the cost of an iterative procedure like Algorithm 3.1 down to  $O(n)$ , provided the number of iterations is problem independent. Here  $n$  is the number of expansion functions, or panels, used in the weighted residuals discretization of the problem.

### 3.2.3 Maintaining Efficiency with the Adaptive Multipole Algorithm

A closer examination of the basic operations of the fast multipole algorithm described in the previous section reveals that the algorithm is a linear operator, effectively approximating  $P$  with some nearby matrix. The details of the approximation lead directly to an new adaptive algorithm appropriate for the sparse charge distributions arising in capacitance calculations.

### The Fast Multipole Algorithm in Matrix Form

The basic operations of the fast multipole algorithm are summarized in Figure 3-6, using two dimensions for clarity. The figure illustrates the calculations required to include the distant panels inside cube 2 in the potential evaluations for the collocation points inside cube 1. The smaller cubes are assumed to be on the lowest level,  $D$ .

During the upward pass, multipole expansions are calculated for the panels in cube 2. First the panels are converted to child cube multipole expansions (Q2M) and then the child expansions are shifted and combined to give cube 2's multipole expansion (M2M). There would be additional M2M conversions if cube 2's children were not on the lowest level.

During the downward pass, cube 2 appears in cube 1's interaction list so cube 2's multipole expansion contributes to cube 1's local expansion (M2L). Other members of cube 1's interaction list, not pictured here, also contribute to cube 1's local expansion via M2L conversions. The

---

<sup>8</sup>Section 4.1.2 gives a precise definition of neighbor cubes for the three-dimensional algorithm.

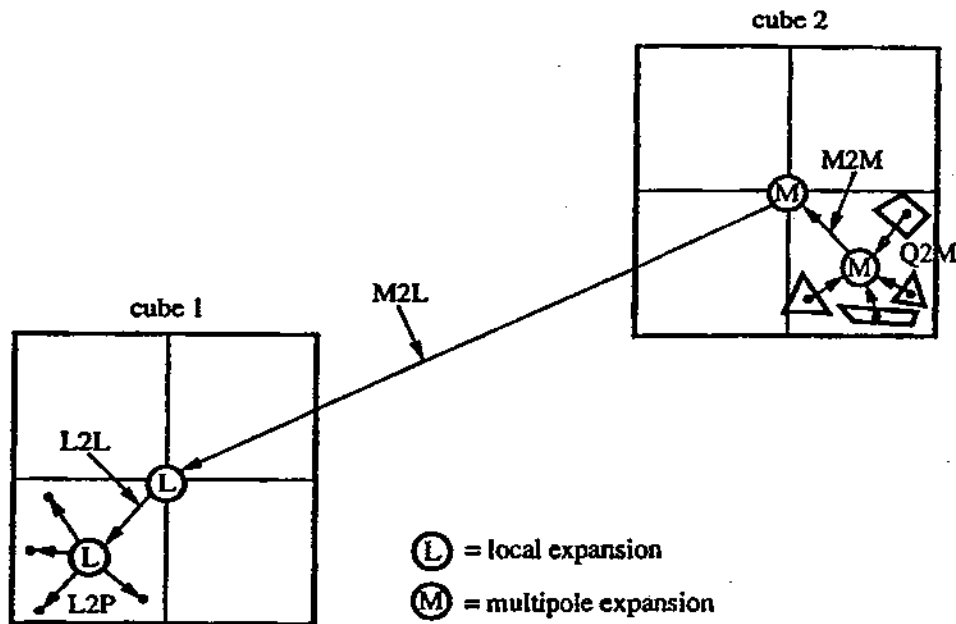


FIGURE 3-6: The basic operations of the fast multipole algorithm.

cube 1 local expansion is shifted down to each of its eight child cubes (L2L) where it becomes the basis of their local expansions. In general, each child cube's local expansion is also influenced by its interaction list cubes via M2L conversions. If cube 1's children were not on the lowest level, then at least one more series of L2L and M2L transformations would be required.

The evaluation pass evaluates the local expansions of cube 1's children at their collocation points (L2P). Also the effects of nearest neighbor and second nearest neighbor cubes are included directly in the child cube potentials (Q2P, not pictured in Figure 3-6). The extension of the neighbor region to include second nearest neighbors is required in three dimensions to maintain accuracy [52].

The formulas for these conversions, assuming the distant panels are approximated by point charges, are given in Appendix B. Only the Q2M formula changes if the point charge approximation is not made. In either case all the formulas are linear transformations. For example, the multipole coefficients given by the Q2M formula are each linear functions of the panel charges. When the multipole and local expansion order is one, that is  $l = 1$  in (3.6) and (3.10), the multipole and local expansions have four real<sup>9</sup> expansion coefficients, and the coefficients of a lowest-level multipole expansion are related to the corresponding lowest-level cube's panel

<sup>9</sup>As explained in Appendix B, the complex coefficients in (3.6) can be combined to give an equivalent expansion in terms of real coefficients.

charges, say  $q_i$ ,  $i = 1, 2, \dots, 6$ , by the linear transformation

$$\begin{bmatrix} & 4 \times 6 \\ & \text{Q2M} \\ & \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \begin{bmatrix} \bar{M}_0^0 \\ \bar{M}_1^0 \\ \bar{M}_1^1 \\ \bar{M}_1^1 \end{bmatrix}.$$

Matrices can be constructed for all the multipole algorithm transformations in a similar fashion.

The matrix representation provides a convenient way of describing how parts of the  $Pq$  product are approximated with the multipole algorithm. The basic multipole algorithm operations illustrated in Figure 3-6 can be written as a chain of matrix-vector products. For example, the transformation from six panel charges,  $q_i$ ,  $i = 1, 2, \dots, 6$ , in a lowest-level cube inside cube 2 to say seven collocation-point potentials,  $f_i$ ,  $i = 10, 11, \dots, 16$ , in a lowest-level cube inside cube 1 using order  $l = 1$  expansions has the form

$$\begin{bmatrix} 7 \times 4 \\ \text{L2P} \end{bmatrix} \begin{bmatrix} 4 \times 4 \\ \text{L2L} \end{bmatrix} \cdots \begin{bmatrix} 4 \times 4 \\ \text{L2L} \end{bmatrix} \begin{bmatrix} 4 \times 4 \\ \text{M2L} \end{bmatrix} \begin{bmatrix} 4 \times 4 \\ \text{M2M} \end{bmatrix} \cdots \begin{bmatrix} 4 \times 4 \\ \text{M2M} \end{bmatrix} \begin{bmatrix} 4 \times 6 \\ \text{Q2M} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_6 \end{bmatrix} = \begin{bmatrix} f_{10} \\ f_{11} \\ \vdots \\ f_{16} \end{bmatrix}. \quad (3.12)$$

When calculating  $Pq$  products using the multipole algorithm, part of the inner products in rows 10-16 of  $Pq$ ,

$$\begin{bmatrix} 7 \times 6 \\ P_{ij} \\ 10 \leq i \leq 16 \\ 1 \leq j \leq 6 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_6 \end{bmatrix} = \begin{bmatrix} f_{10} \\ f_{11} \\ \vdots \\ f_{16} \end{bmatrix}. \quad (3.13)$$

is replaced by the sequence of multipole matrix-vector products (3.12).

Since (3.12) requires more work than (3.13), it is tempting to conclude that the overall multipole computation is less efficient than the complete, explicit  $Pq$  product. The multipole evaluation (3.12) requires at least the Q2M, M2L and L2P operations, leading to  $7 \times 4 + 4 \times 4 + 4 \times 6 = 68$  multiply-adds in this example, more than the  $7 \times 6 = 42$  required for the corresponding part of the  $Pq$  product (3.13). However, most of the intermediate results calculated in a particular multipole evaluation like (3.12) are reused to approximate other parts of the  $Pq$  product (see Section 3.2.2), usually making the overall multipole calculation more efficient than the complete  $Pq$  product.

Since all the transformation matrices are functions only of geometry, a large amount of the overhead associated with the multipole algorithm can be eliminated by calculating the

transformation matrices once for a particular problem, as in the implementation described in Section 4.1. The result is effectively the multipole algorithm's approximation to  $P$  represented as a collection of transformation matrices. Recalculating the transformation matrices for every approximate  $Pq$  product requires roughly an order of magnitude more computation than reusing a stored approximation to  $P$ .

### Multipole Algorithm Efficiency

The multipole approximation is inefficient for problems that result in cubes with small numbers of charge panels,  $n_q$ , or collocation points,  $n_p$ . In particular, if  $n_q \leq (l+1)^2$  or  $n_p \leq (l+1)^2$  then order  $l$  expansions, with  $(l+1)^2$ -entry coefficient vectors (see also Appendix B), make the algorithm unnecessarily expensive and inaccurate. The matrix notation is useful for illustrating this.

Figure 3-7a illustrates the calculation of the multipole expansion  $M$ , using two dimensions for clarity. In this example the expansions are order  $l = 1$ , making the lowest-level cube multipole expansion  $M_b$  a vector with  $(l+1)^2 = 4$  entries calculated by multiplying a Q2M matrix by the corresponding three-element panel-charge vector,

$$\begin{matrix} 4 \times 3 & 3 \times 1 & 4 \times 1 \\ \left[ \begin{array}{c} \text{Q2M}_b \end{array} \right] & \left[ \begin{array}{c} q \end{array} \right] & = & \left[ \begin{array}{c} M_b \end{array} \right], \end{matrix}$$

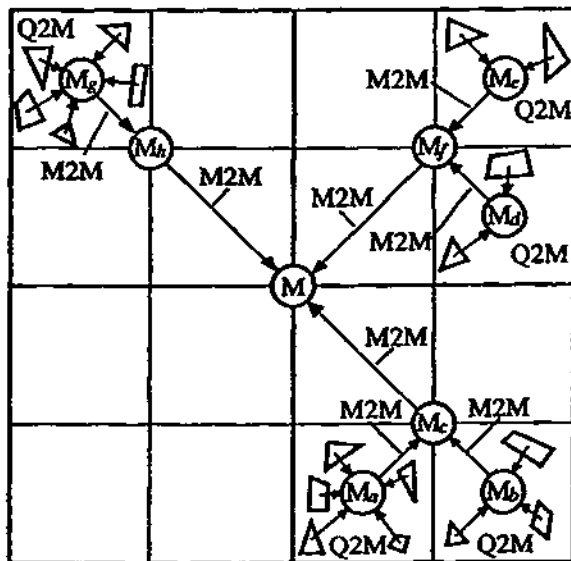
so that the panel-charge vector is represented by a longer multipole vector. The multipole vector  $M_b$  then contributes to the parent multipole expansion  $M_c$  (M2M) and, during the downward pass, to local expansions for cubes that contain  $M_b$ 's cube in their interaction lists. (M2L). Alternatively, the multipole expansion can be replaced by the shorter charge vector as in Figure 3-7b. The panel-charge vector contributes directly to  $M_c$  using a  $4 \times 3$  Q2M matrix, eliminating the M2M operation used in Figure 3-7a. Further savings result from direct panel-charge conversions to local expansions using  $3 \times 4$  Q2L matrices (see Appendix B), rather than  $4 \times 4$  M2L transformations, during the downward pass. Since the approximation  $M_b$  is not used, the Figure 3-7b strategy is not only more efficient, but also more accurate, than the non-adaptive method of Figure 3-7a.

In general, the  $n_q$ -entry panel-charge vector for any cube on any level is a more accurate and at least as efficient representation of the panel-charge potential if there are no more panel charges than expansion coefficients,

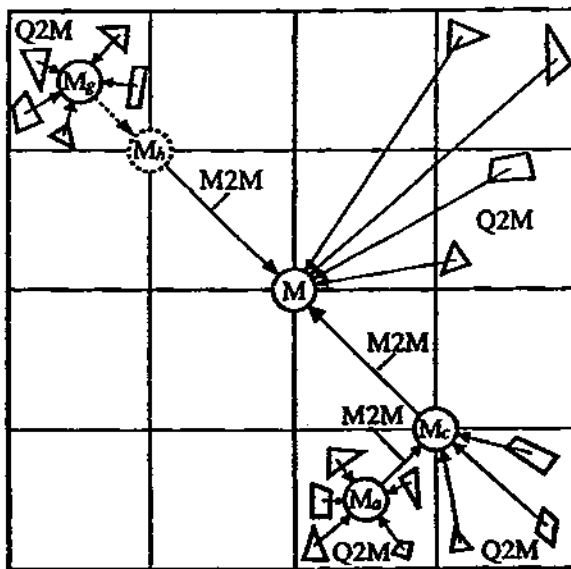
$$n_q \leq (l+1)^2. \quad (3.14)$$

The cube associated with  $M_f$  in Figure 3-7a is an example of a cube that is not on the lowest level but is still best represented by its panel-charge vector, as in Figure 3-7b.





(a)



(b)

FIGURE 3-7: The multipole-expansion hierarchies constructed by the non-adaptive (a) and adaptive (b) multipole algorithms for an example panel-charge distribution.

Figure 3-7 also illustrates another instance where it is inefficient to compute multipole expansions. The expansion  $M_h$  in Figure 3-7a corresponds to a cube with a single non-empty lowest-level child with expansion  $M_g$ . Since the  $M_h$  cube has only one non-empty child cube,  $M_h$  represents the same panel charge as  $M_g$ , but is less accurate due to the M2M operation used to shift  $M_g$  to  $M_h$ 's center. If rather than calculating  $M_h$ , a copy of  $M_g$  is used for  $M_h$ , as in Figure 3-7b, then the resulting calculation is more accurate and efficient. In general, any number of cubes with single, non-empty child cubes, each one the child of the previous one, require only one multipole-expansion calculation.

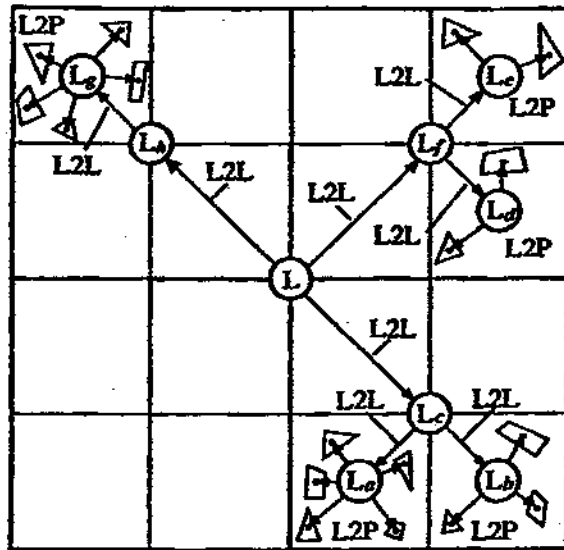
Analogous optimizations are possible for the local expansion calculations. In general a cube's  $n_p$  evaluation-point potentials are represented at least as efficiently, and more accurately, by directly evaluating the multipole-expansion and panel-charge potential contributions using M2P and Q2P transformations rather than building a local expansion using M2L and Q2L transformations if there are no more evaluation points than local expansion coefficients,

$$n_p \leq (l+1)^2. \quad (3.15)$$

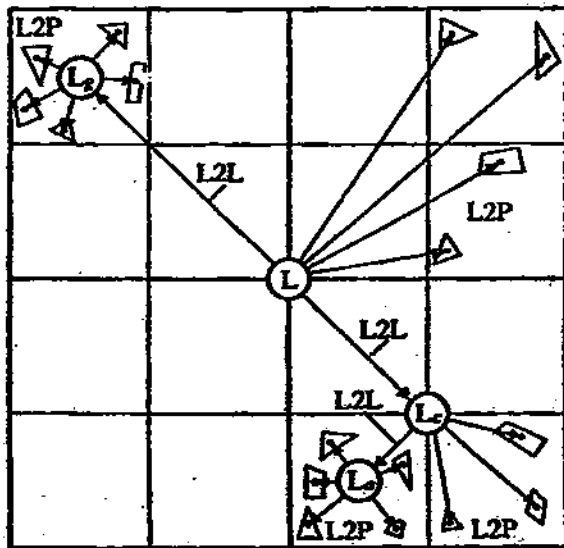
For a cube that satisfies (3.15), building a local expansion requires at least as much computation as directly evaluating the potential at all the cube's evaluation points. In particular, the local expansion build followed by its evaluation with an  $n_p \times (l+1)^2$  L2P transformation has the form

$$\begin{aligned} & \begin{bmatrix} n_p \times (l+1)^2 \\ \text{L2P} \end{bmatrix} \left( \begin{bmatrix} (l+1)^2 \times (l+1)^2 \\ \text{L2L} \end{bmatrix} \begin{bmatrix} L_p \end{bmatrix} \right) \\ & + \begin{bmatrix} (l+1)^2 \times n_1 \\ \text{M2L} \\ \text{or} \\ \text{Q2L} \end{bmatrix} \begin{bmatrix} M_1 \\ \text{or} \\ q_1 \end{bmatrix} + \dots + \begin{bmatrix} (l+1)^2 \times n_l \\ \text{M2L} \\ \text{or} \\ \text{Q2L} \end{bmatrix} \begin{bmatrix} M_l \\ \text{or} \\ q_l \end{bmatrix} \Bigg) = \begin{bmatrix} n_p \times 1 \\ p \end{bmatrix}. \quad (3.16) \end{aligned}$$

Here the cube's evaluation-point potential vector is  $p$  and its interaction list has  $l$  entries, the  $i$ -th entry corresponding to an  $(l+1)^2 \times n_i$  M2L or Q2L transformation matrix. The terms in parenthesis add to give the cube's local expansion. As mentioned in the discussion of Figure 3-7, Q2L matrices with  $(l+1)^2$  or less columns are used to include the effects of charge in distant cubes that lack sufficient panels to warrant a multipole expansion. The L2L transformation term is present only if the cube's parent has a local expansion,  $L_p$ . The corresponding direct evaluation involves analogous, but no bigger, M2P, Q2P and, if there is a parent local expansion,



(a)



(b)

FIGURE 3-8: The local expansion hierarchies constructed by the non-adaptive (a) and adaptive (b) multipole algorithms for an example panel-charge distribution.

L2P transformation matrices,

$$\begin{aligned}
 & n_p \times (l+1)^2 \\
 & \left[ \begin{array}{c} \text{L2P} \end{array} \right] \left[ \begin{array}{c} L_p \end{array} \right] \\
 & + \left[ \begin{array}{c} n_p \times n_1 \\ \text{M2P or} \\ \text{Q2P} \end{array} \right] \left[ \begin{array}{c} M_1 \\ \text{or} \\ q_1 \end{array} \right] + \dots + \left[ \begin{array}{c} n_p \times n_l \\ \text{M2P or} \\ \text{Q2P} \end{array} \right] \left[ \begin{array}{c} M_l \\ \text{or} \\ q_l \end{array} \right] = \left[ \begin{array}{c} n_p \times 1 \\ p \end{array} \right], \quad (3.17)
 \end{aligned}$$

making the direct evaluation require the same or less computation as the local expansion calculation. Since the direct evaluation eliminates at least one local-expansion approximation, it is also more accurate.

Figure 3-8 illustrates the evaluation point distribution resulting from the geometry of Figure 3-7, assuming panel-centroid evaluation points. In a manner analogous to the Figure 3-7 optimization, unnecessary local expansions are removed from Figure 3-8a when (3.15) holds, giving Figure 3-8b.

Since local expansions are approximations for the potential due to distant panel charge, the efficient treatment of cubes with single children is slightly different than in the upward pass case. A multipole expansion depends on nearby charge (Q2M), or nearby charge represented by the corresponding child cubes' multipole expansions (M2M), while a local expansion depends only on distant charge represented by its interaction list (M2L or Q2L) and the corresponding parent cube's local expansion (L2L). Thus the charge represented by the local expansion for an only-child cube can differ from that represented by the parent cube's local-expansion due to the only child's interaction-list contributions. In contrast, the only-child cube's multipole expansion can always be used in the place of its parent cube's expansion, since the only-child cube's charge is the only charge in its parent cube. This reasoning appears to indicate that a local expansion like  $L_h$  in Figure 3-8a always needs to be calculated since it generally cannot be replaced by its parent local, in this case  $L$ . However,  $L_h$  exists solely to summarize the effects of  $L$  and the  $L_h$  cube's interaction list entries, and that information is used only to form the only-child expansion  $L_g$ . It is therefore possible to append the  $L_h$  cube's interaction list to the  $L_g$  cube's interaction list, and eliminate the intermediate  $L_h$  calculation. The resulting computation is as in Figure 3-8b, where  $L_g$  is calculated directly from  $L$  (L2L) and the merged  $L_g$  and  $L_h$  interaction lists (M2L or Q2L). In contrast to the the upward pass case, no local expansion or local expansion copy is maintained in place of  $L_h$ . The same interaction-list merging is done for the lowest-level cube direct evaluations in Figure 3-8b that replace the local expansions  $L_d$ ,  $L_e$  and  $L_f$  in Figure 3-8a. The interaction-list entries for the  $L_f$  cube are appended to the  $L_d$  and  $L_e$  cube lists for use in the direct evaluations of Figure 3-8b.

In general, all the operations in Figure 3-8 require interaction-list evaluations, except for the lowest-level local expansion evaluations. Depending on the interaction lists, it may be possible to replace some of the local expansions in Figure 3-8b by more efficient direct evaluations. Such a replacement is possible because although a direct evaluation is more efficient when (3.15) holds, violating (3.15) does not necessarily make using a local expansion less costly than direct evaluation. The final cost of the local expansion calculation may be greater since the local expansion requires evaluation, possibly through a hierarchy of lower-level local expansions, while the direct evaluation gives the evaluation-point potentials without further computation. The potential savings, however, are not enough to warrant checking for this possibility, as Appendix D demonstrates.

### The Adaptive Fast Multipole Algorithm

The simple fast multipole algorithm illustrated in Figures 3-7a and 3-8a builds multipole and local expansions for every lowest-level cube, regardless of their efficiency. Following the more efficient strategies of Figures 3-7b and 3-8b leads to the adaptive fast multipole algorithm. Any cube that satisfies (3.14) has at least as many multipole-expansion coefficients as panel charges and is called exact for the multipole expansion. Cubes that satisfy (3.15) have at least as many local-expansion coefficients as evaluation points and are called exact for the local expansion. The adaptive fast multipole algorithm constructs no expansions of the corresponding type for exact cubes. It also eliminates the extra expansion calculations associated with cubes that are only children and have only one child themselves, as illustrated in Figures 3-7b and 3-8b.

Collecting these ideas leads to the adaptive fast multipole algorithm, stated briefly below. Section 4.1 describes the details of the particular implementation used here.

**Algorithm 3.3 (Adaptive Fast Multipole Algorithm)** *To calculate an approximate  $Pq$  product,*

1. *If this is the first  $Pq$  for a new problem, choose an expansion order  $l$  and compute appropriate transformation matrices for all cubes, taking exact cubes into account, and store for subsequent calls. Included here are  $Q2P$  matrices representing interactions between panels in the neighbor cube regions for all the lowest-level cubes. If matrices are already calculated, go to the next step.*
2. *Perform the upward pass. Apply  $Q2M$  and  $M2M$  matrices to obtain multipole expansions in all non-exact cubes.*
3. *Perform the downward pass. Apply  $M2L$ ,  $L2L$  and  $Q2L$  matrices to obtain local expansions in all non-exact cubes.*
4. *Perform the evaluation pass. Apply  $L2P$ ,  $M2P$  and  $Q2P$  matrices to obtain all collocation point potentials. Included here are products with the  $Q2P$  matrices representing evalua-*

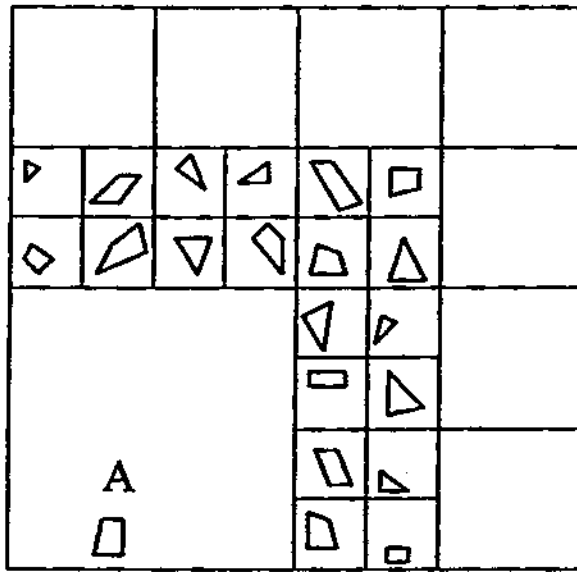
*tions of potentials due to panels in the neighbor cube regions for all the lowest-level cubes.*

The original adaptive fast multipole algorithm [52] relies on an irregular partitioning that ensures that all lowest level cubes, which need not be the same size, contain roughly the same number of panels and evaluation points (see Figure 3-9a). The exact cubes of the adaptive method presented here (Figure 3-9b), considered as coarsely as possible, correspond roughly to the irregular, lowest-level cubes of the original algorithm. The principle advantage of the original method appears to be that it constructs multipole and local expansions for every cube, thus preserving the hierarchical expansion structure of the simple multipole method and eliminating the need for Q2L and M2P transformations. The variable size of the lowest level cubes, however, makes it difficult to determine a given cube's neighboring cubes. This complicates the nearby panel-potential evaluations. The technique presented here uses the same regular partitioning of the simple method, leading to an easier implementation and a consistent treatment of nearby panel potentials.

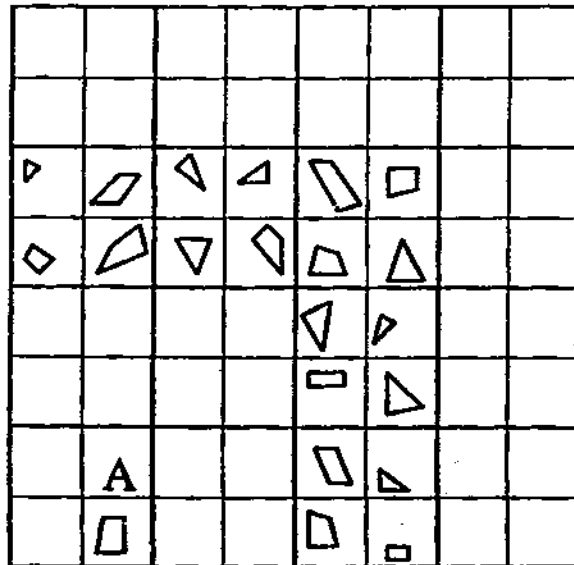
The example in Figure 3-9a illustrates the inefficiency of the nonuniform problem-domain partitioning. The problem-domain partitioning is halted when each child-square has no more than one panel. Computing the potential at the centroid of panel A using a multipole algorithm based on this nonuniform partitioning requires nine direct potential evaluations, for the nine panels in nearest-neighbor squares bordering the square containing A, and eleven multipole evaluations for the eleven nonempty squares not bordering A. The result is more work than in the supposedly less efficient non-adaptive algorithm, which allows the evaluation of five multipole expansions, each including the effects of four panels, to compute the potential at A. By preserving the uniform partitioning used by the non-adaptive algorithm, the adaptive algorithm used here, illustrated in Figure 3-9b, allows the use of the same five multipole evaluations. In fact, using the uniform partitioning for the adaptive algorithm ensures that cubes' neighbors never fragment nearby charge cubes (as in Figure 3-9a), making the adaptive algorithm of Figure 3-9b always at least as efficient as the corresponding non-adaptive algorithm.

### 3.2.4 Preconditioning the Multipole Accelerated Iterative Method

Combining the ideas of Secions 3.2.1 and 3.2.2, the capacitance solution method of this thesis uses a preconditioned GMRES iterative loop, with iterates calculated with the adaptive multipole algorithm. Using a preconditioner applies the iterative method to  $\tilde{C}Pq = \tilde{C}f$  rather than  $Pq = f$  and minimizes variations in iteration counts across different problems. The actual structure of the preconditioning matrix,  $\tilde{C}$ , is dictated by the convergence properties of GMRES and the structure of the multipole algorithm.



(a)



(b)

FIGURE 3-9: Partitionings used in Algorithm 3.3, (b), and by the adaptive algorithm of [19], (a).

## GMRES Convergence Properties

The choice of preconditioner matrix depends on GMRES's convergence properties, which are determined by how GMRES calculates the next iterate (step 3 in Algorithm 3.1) [112]. For the  $n \times n$  system  $Pq = f$ , the new iterate  $q^{i+1}$  is

$$q^{i+1} = q^0 + z^{i+1}, \quad i = 0, 1, \dots, n-2, \quad (3.18)$$

where  $q^0$  is the initial guess giving initial residual  $r^0 = f - Pq^0$ , and  $z^{i+1}$  is a vector in the span of the Krylov vectors

$$r^0, Pr^0, P^2r^0, \dots, P^i r^0$$

such that the new residual

$$r^{i+1} \triangleq f - Pq^{i+1} = r^0 - Pz^{i+1}, \quad i = 0, 1, \dots, n-2, \quad (3.19)$$

has minimal Euclidean norm. In particular, on the first iteration  $z^1$  is a vector in the space spanned by the single vector  $r^0$ , so

$$z^1 = \alpha_1^1 r^0,$$

with  $\alpha_1^1$  chosen so that

$$\|r^1\|_2 = \|r^0 - Pz^1\|_2 = \|(I - \alpha_1^1 P)r^0\|_2 \quad (3.20)$$

$$= \|g^1(P)r^0\|_2 \quad (3.21)$$

is minimized. Here  $g^1(x) \triangleq 1 - \alpha_1^1 x$ , with the understanding that the leading 1 changes to an identity matrix when the scalar argument  $x$  is replaced with a matrix. In general, the  $(i+1)$ -th iteration chooses  $z^{i+1}$  from all the vectors in the space spanned by the vectors  $r^0, Pr^0, P^2r^0, \dots, P^i r^0$ , so

$$z^{i+1} = \sum_{j=0}^i \alpha_j^{i+1} P^j r^0,$$

with the  $\alpha_j^{i+1}$  coefficients chosen to minimize the norm of the residual,

$$\begin{aligned} \|r^{i+1}\|_2 &= \|r^0 - Pz^{i+1}\|_2; \\ &= \|(I - \sum_{j=0}^i \alpha_j^{i+1} P^{j+1})r^0\|_2; \end{aligned} \quad (3.22)$$

$$= \|g^{i+1}(P)r^0\|_2, \quad (3.23)$$

where  $g^{i+1}(x) \triangleq 1 - \sum_{j=0}^i \alpha_j^{i+1} x^j$ , and  $I$  is the identity matrix.

The polynomial structure of the term in parenthesis in (3.22) is the key to linking GMRES's residual-minimization properties to the eigen-structure of  $P$ . The minimization problem that determines  $z^{i+1}$  can be restated as the problem of finding a degree  $i+1$  polynomial  $g^{i+1}(x)$  with  $g^{i+1}(0) = 1$  such that

$$\|r^{i+1}\|_2 = \|g^{i+1}(P)r^0\|_2 \quad (3.24)$$



is minimal. The use of the polynomial automatically constrains  $z^{i+1}$  to be in the correct Krylov subspace.

Using the polynomial interpretation, it is possible to say how the structure of  $P$  affects the efficiency of the residual minimization performed on each iteration [131, 74]. If  $P$  is diagonalizable then

$$P = VDV^{-1}, \quad (3.25)$$

where  $D$  is a diagonal matrix with  $P$ 's eigenvalues on the diagonal, and  $V$  is a matrix whose columns are  $P$ 's eigenvectors. Substitution in (3.24) gives

$$\|r^{i+1}\|_2 = \|g^{i+1}(VDV^{-1})r^0\|_2,$$

so that

$$\begin{aligned} \|r^{i+1}\|_2 &\leq \|V\|_2 \|V^{-1}\|_2 \|g^{i+1}(D)\|_2 \|r^0\|_2; \\ \frac{\|r^{i+1}\|_2}{\|r^0\|_2} &\leq \kappa(V) \|g^{i+1}(D)\|_2, \end{aligned} \quad (3.26)$$

where

$$\kappa(V) \triangleq \|V\|_2 \|V^{-1}\|_2 \quad (3.27)$$

is the condition number of  $P$ 's eigenvector matrix  $V$  [50]. For the matrices arising from capacitance-problem discretizations,  $\kappa(V)$  is reasonably close to one and  $\|g^{i+1}(D)\|_2$  closely tracks the residual norm. Because of  $D$ 's structure,  $\|g^{i+1}(D)\|_2$  is equivalent to the maximum magnitude attained by  $g^{i+1}(\lambda)$  when  $\lambda$  is one of  $P$ 's eigenvalues. Thus GMRES iterations for  $Pq = f$  produce polynomials  $g^{i+1}(x)$  that minimize the maximum magnitude attained by  $g^{i+1}(\lambda)$ , making  $g^{i+1}(x)$  an  $(i+1)$ -th degree interpolant to  $P$ 's eigenvalues with the normalization constraint  $g^{i+1}(0) = 1$ .

In fact, GMRES chooses  $g^{i+1}$  to minimize  $\|g^{i+1}(D)r^0\|_2$ , not  $\|g^{i+1}(D)\|_2 \|r^0\|_2$ , so  $g^{i+1}$  really depends on the initial guess and right-hand-side vector through  $r^0$ . In practice differences in the initial residuals only effect the first few iterations, so viewing the polynomial construction as an eigenvalue interpolation is a good approximation that is useful for analyzing convergence.

To illustrate this, consider the seven-panel example pictured in Figure 3-10 with corresponding potential coefficient matrix

$$P = \begin{bmatrix} 3.5255 & 0.9286 & 0.4900 & 0.3303 & 0.2487 & 0.1993 & 0.1663 \\ 0.9286 & 3.5255 & 0.9286 & 0.4900 & 0.3303 & 0.2487 & 0.1993 \\ 0.4900 & 0.9286 & 3.5255 & 0.9286 & 0.4900 & 0.3303 & 0.2487 \\ 0.3303 & 0.4900 & 0.9286 & 3.5255 & 0.9286 & 0.4900 & 0.3303 \\ 0.2487 & 0.3303 & 0.4900 & 0.9286 & 3.5255 & 0.9286 & 0.4900 \\ 0.1993 & 0.2487 & 0.3303 & 0.4900 & 0.9286 & 3.5255 & 0.9286 \\ 0.1663 & 0.1993 & 0.2487 & 0.3303 & 0.4900 & 0.9286 & 3.5255 \end{bmatrix}, \quad (3.28)$$

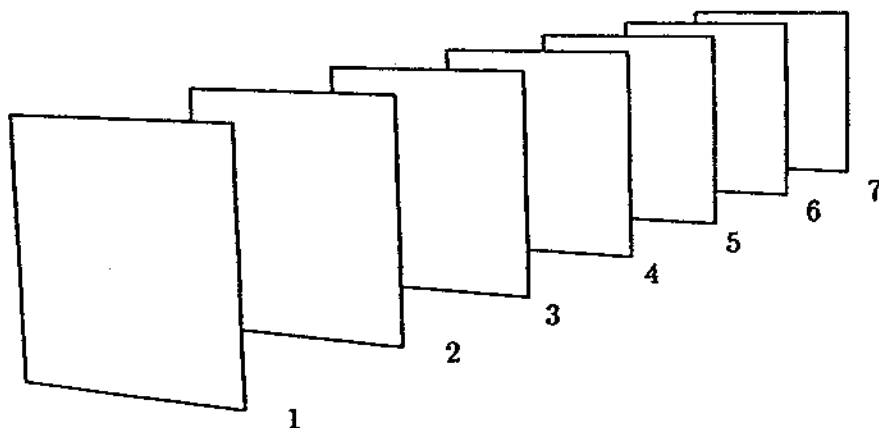


FIGURE 3-10: Seven 1m×1m parallel panels spaced 1m apart.

$Pq = f, \kappa(V) = 1.0$			$\tilde{C}Pq = \tilde{C}f, \kappa(V) = 1.1$		
$i + 1$	$\ r^{i+1}\ _2 / \ r^0\ _2$	$\ g^{i+1}(D)\ _2$	$i + 1$	$\ r^{i+1}\ _2 / \ r^0\ _2$	$\ g^{i+1}(D)\ _2$
1	0.31204	0.69797	1	0.08301	0.21784
2	0.08721	0.15714	2	0.00916	0.05283
3	0.01667	0.03994	3	0.00059	0.00122
4	0.00237	0.00709	4	0.00003	0.00006

(a)

(b)

Table 3-1: GMRES polynomials for  $Pq = f$  (a) and  $\tilde{C}Pq = \tilde{C}f$  (b), with  $P$  and  $\tilde{C}$  given by (3.28) and (3.31) respectively, and where  $f$  is the first column of the identity matrix.

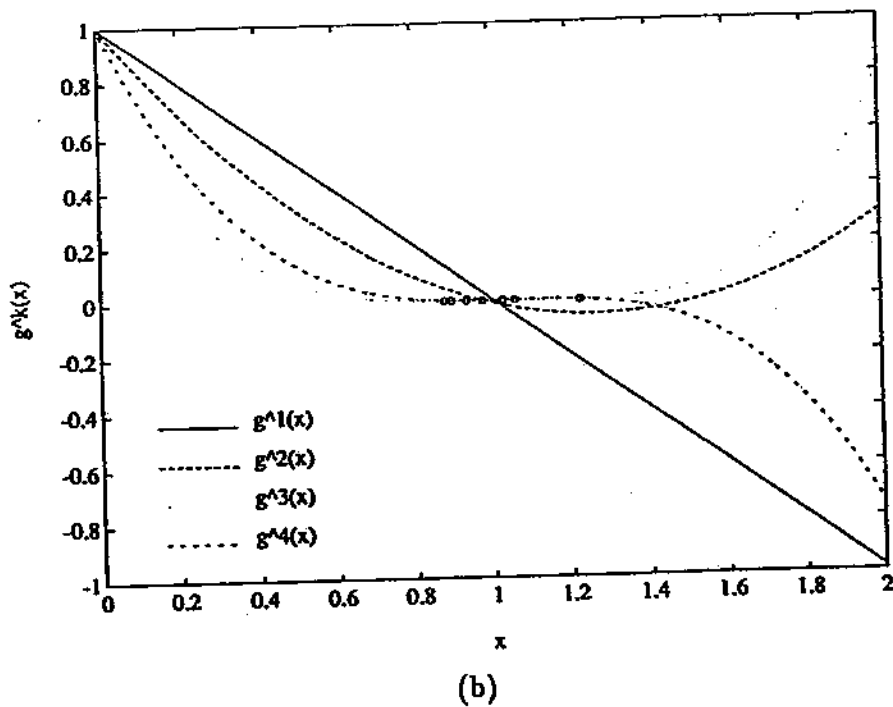
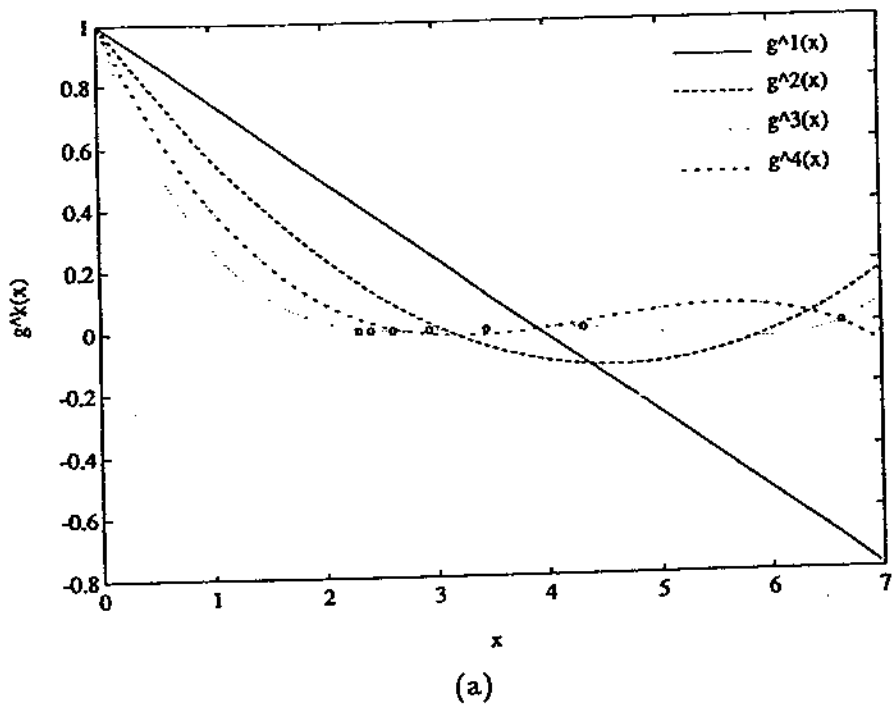


FIGURE 3-11: GMRES polynomials for  $Pq = f$  (a) and  $\tilde{C}Pq = \tilde{C}f$  (b), with  $P$  and  $\tilde{C}$  given by (3.28) and (3.31) respectively, and where  $f$  is the first column of the identity matrix.

calculated using (3.4). Figure 3-11a and Table 3-1a give the polynomials, the bounds they generate, and the normalized residuals for the first four iterations of GMRES applied to  $Pq = f$ , with  $f$  taken as the first column of the identity matrix. Figure 3-11a shows how the GMRES polynomial is a progressively better interpolant of  $P$ 's eigenvalues, represented by circles on the  $x$  axis in the figure<sup>10</sup>. This fact leads to smaller values of the bound  $\|g^{i+1}(D)\|_2$  on each iteration, which track the normalized residual (see Table 3-1a). Thus a decrease in the residual goes hand in hand with a decrease in  $\|g^{i+1}(D)\|_2$ , which in turn is caused by a good fit of the current GMRES polynomial to  $P$ 's eigenvalues. The convergence of GMRES is therefore strongly affected by the placement of  $P$ 's eigenvalues. In the Figure 3-11a example, the GMRES polynomials interpolate the lone eigenvalue near 6.5 almost exactly by the third iteration, while the cluster of eigenvalues closer to the origin, because of the normalization constraint  $g^{i+1}(0) = 1$ , are not as well approximated initially. In general the interpolation matches the eigenvalues in a particular eigenvalue cluster with comparable error, usually with better matching for clusters farther from the origin. Thus eigenvalues of  $P$  that are spread out complicate the interpolation and lead to slow convergence, while clustered eigenvalues lead to rapid convergence.

### GMRES Preconditioning for the Capacitance Problem

Applying GMRES to  $\tilde{C}Pq = \tilde{C}f$  rather than  $Pq = f$  speeds convergence if  $\tilde{C}P$ 's eigenvalues are easier to interpolate than  $P$ 's. If the preconditioner matrix  $\tilde{C} = P^{-1}$ , then  $\tilde{C}P = I$  has all its eigenvalues clustered at one, making the GMRES interpolation exact for all eigenvalues on the first iteration. Although calculating  $P^{-1}$  for the preconditioner is prohibitively expensive, a good general strategy is to choose  $\tilde{C}$  to be a low-cost approximation to  $P^{-1}$ , so as to cluster  $\tilde{C}P$ 's eigenvalues around one. For the capacitance problem,  $P$ 's structure ensures that easily computed  $\tilde{C}$ 's, derived using the inverse-approximation strategy, give preconditioned matrices  $P\tilde{C}$  with well-clustered eigenvalues.

Although preconditioning generally speeds convergence, it also ensures convergence to an accurate solution. When solving  $Pq = f$ , the object is to minimize the error in the current iterate,  $q - q^{i+1}$ , where  $q$  is the exact solution. GMRES minimizes the residual on each iteration, which is related to the solution error through  $P^{-1}$ ,

$$\begin{aligned} q - q^{i+1} &= P^{-1}f - q^{i+1}; \\ &= P^{-1}(f - Pq^{i+1}); \\ q - q^{i+1} &= P^{-1}r^{i+1}. \end{aligned}$$

Taking the norm of this equation and combining it with the norm of  $Pq = f$  gives a relative

<sup>10</sup>If  $P$ 's eigenvalues are complex, the polynomials can still be interpreted as interpolants, but the effect is more difficult to illustrate in a two-dimensional plot.

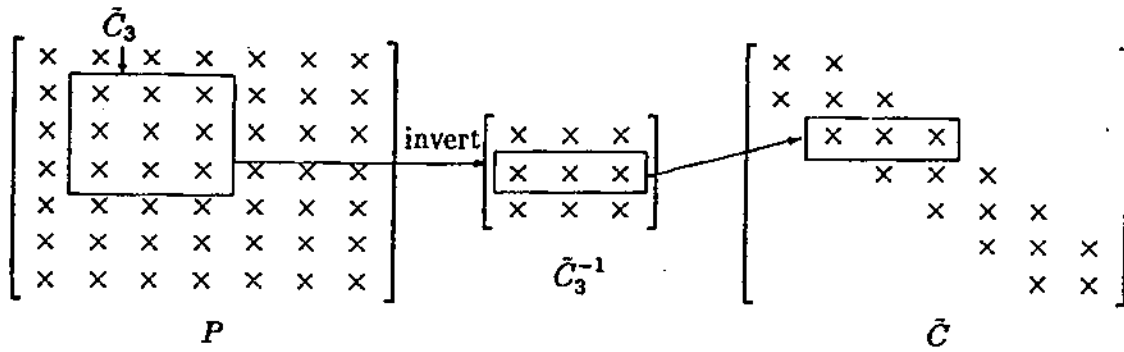


FIGURE 3-12: The steps leading to the third row of the preconditioner  $\tilde{C}$  for the example  $P$  (“ $\times$ ” denotes a non-zero element).

error bound in terms of the residual,

$$\begin{aligned} \frac{\|q - q^{i+1}\|_2}{\|q\|_2} &\leq \|P\|_2 \|P^{-1}\|_2 \frac{\|r^{i+1}\|_2}{\|f\|_2}; \\ &\leq \kappa(P) \frac{\|r^{i+1}\|_2}{\|f\|_2}, \end{aligned} \quad (3.29)$$

where  $\kappa(P)$  is the condition number of  $P$ . The analogous bound for the preconditioned system  $\tilde{C}Pq = \tilde{C}f$  is

$$\frac{\|q - q^{i+1}\|_2}{\|q\|_2} \leq \kappa(\tilde{C}P) \frac{\|r^{i+1}\|_2}{\|f\|_2}, \quad (3.30)$$

where  $r^{i+1}$  is now the residual of the preconditioned system. The bound (3.29) suggests that if  $P$  is poorly conditioned then a small normalized residual does not necessarily imply that the solution error is commensurately small. Thus preconditioning  $P$  so that the effective system matrix has its eigenvalues clustered near one, leading to a condition number  $\kappa(\tilde{C}P) \approx 1$  in (3.30), not only leads to faster GMRES convergence, but also ensures that the residual is a good measure of the solution error.

The potential-coefficient matrix entry  $P_{ij}$  is the potential at panel  $i$  due to a unit charge on panel  $j$  (see also Section 3.1). Since  $P_{ij}$  falls off like the inverse of the distance between panels  $i$  and  $j$ , the potential coefficients corresponding to distant panel pairs are relatively small. A good preconditioner depends on the larger  $P_{ij}$ 's, thereby including the stronger nearby interactions, and ignores the distant interactions to save computation.

The one-dimensional nature of the seven-panel problem of Figure 3-10, with  $P$  given by (3.28), makes such a strategy easy to implement. Suppose that any panel within 1m of a given panel is considered nearby. Panel 3, for example, is near panels 2 and 4. The potential coefficients corresponding to the interactions between a given panel and its three nearby panels form a  $3 \times 3$  (panels 2–6) or  $2 \times 2$  (panels 1 and 7) principle submatrix in  $P$ . Figure 3-12 shows the position of the submatrix corresponding to panel 3,  $\tilde{C}_3$ . Combining the small submatrix inverses gives a good preconditioner since the explicit inverses of the strongly coupled nearby problems capture the essence of the full inverse  $P^{-1}$ . Since only small submatrices of  $P$  are inverted, the

cost is low. Figure 3-12 illustrates how the panel- $i$  row of the inverted submatrix corresponding to panel  $i$  becomes the non-zero part of the  $i$ -th row in  $\tilde{C}$ , leading to a preconditioner that combines parts of all the nearby interaction submatrices.

The complete preconditioner for the (3.28)  $P$  obtained using this strategy,

$$\tilde{C} = \begin{bmatrix} 0.3048 & -0.0803 & & & & & \\ -0.0747 & 0.3230 & -0.0747 & & & & \\ & -0.0747 & 0.3230 & -0.0747 & & & \\ & & -0.0747 & 0.3230 & -0.0747 & & \\ & & & -0.0747 & 0.3230 & -0.0747 & \\ & & & & -0.0747 & 0.3230 & -0.0747 \\ & & & & & -0.0803 & 0.3048 \end{bmatrix}, \quad (3.31)$$

has non-zero terms that nearly match the entries of the inverse,

$$P^{-1} = \begin{bmatrix} 0.3076 & -0.0732 & -0.0188 & -0.0105 & -0.0072 & -0.0056 & -0.0056 \\ -0.0732 & 0.3250 & -0.0689 & -0.0164 & -0.0090 & -0.0062 & -0.0056 \\ -0.0188 & -0.0689 & 0.3260 & -0.0684 & -0.0162 & -0.0090 & -0.0072 \\ -0.0105 & -0.0164 & -0.0684 & 0.3262 & -0.0684 & -0.0164 & -0.0105 \\ -0.0072 & -0.0090 & -0.0162 & -0.0684 & 0.3260 & -0.0689 & -0.0188 \\ -0.0056 & -0.0062 & -0.0090 & -0.0164 & -0.0689 & 0.3250 & -0.0732 \\ -0.0056 & -0.0056 & -0.0072 & -0.0105 & -0.0188 & -0.0732 & 0.3076 \end{bmatrix}.$$

The results of applying GMRES to the preconditioned system  $\tilde{C}Pq = \tilde{C}f$  given in Figure 3-11b and Table 3-1b are further evidence of  $\tilde{C}$ 's similarity to the inverse. The preconditioned matrix's eigenvalues, plotted as o's in Figure 3-11b, are strongly clustered around one, making the GMRES polynomial interpolation more effective. The more efficient interpolation leads to the faster convergence of the residual norms in Table 3-1b. Although  $\tilde{C}P$  has  $\kappa(V) > 1$ , the bounds on the residuals given in Table 3-1b are still tight enough to show the link between the eigenvalue clustering and the residual norm convergence.

A similar strategy leads to preconditioners for somewhat more general problems than the Figure 3-10 geometry. If each panel in Figure 3-10 is replaced by a cube containing any number of panels, the geometry of Figure 3-13 results. Constructing a preconditioner for the corresponding potential coefficient matrix  $P$  follows the same strategy as for the Figure 3-10 case, except that the cubic regions determine which panels are nearby. In particular, the preconditioner build follows the same steps illustrated in Figure 3-12, with the x's symbols now representing block matrices in  $P$  rather than individual entries. The rows extracted from  $\tilde{C}_i$  are now the rows corresponding to all the panels inside cube  $i$ , and, in this one-dimensional-cube-array case,  $\tilde{C}$  is block-tridiagonal.

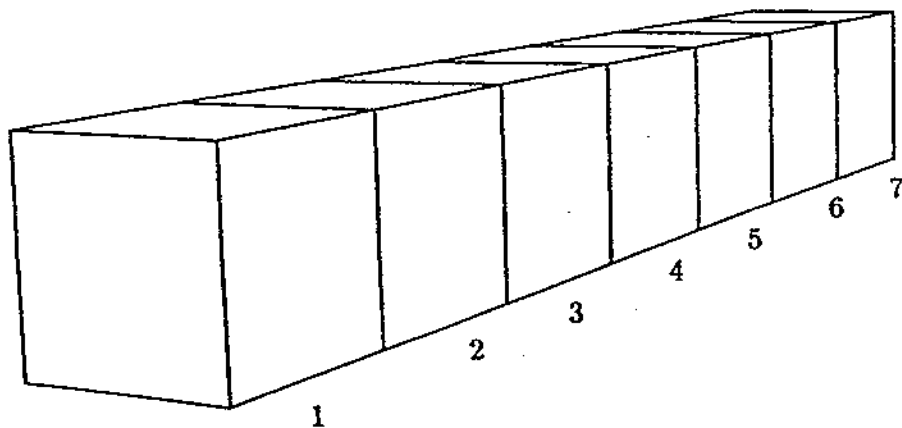


FIGURE 3-13: A one-dimensional cubic partitioning of a region containing panels.

### Constructing a GMRES Preconditioner when Using the Multipole Algorithm

The multipole algorithm recursively divides the largest cube containing all the panels in the problem to obtain a hierarchy of cubic regions (see Section 3.2.2). The nearby interactions between a lowest-level cube and its neighbor cubes are calculated directly from the panel charges and stored potential coefficients. These potential coefficients are the only entries of  $P$  explicitly computed by the multipole algorithm, yet they represent all of the nearby interaction, and are therefore useful for constructing a preconditioner.

A very effective preconditioner that takes advantage of all the potential coefficients already required by the multipole algorithm results when the strategy applied to the one-dimensional cube array in Figure 3-13 is blindly extended to three dimensions by considering all neighbor cubes of a given cube in the multipole algorithm hierarchy to be nearby the given cube. Such an approach is used in a different context in [135].

**Algorithm 3.4 (Multipole-Accelerated GMRES Preconditioner Build)** *To construct a GMRES preconditioner  $\tilde{C}$  for a  $n$ -panel problem using all the entries of  $P$  calculated by the multipole algorithm,*

1. *Set cube  $i$  equal to the first non-empty lowest-level cube and  $\tilde{C} = 0$ .*
2. *Look up the previously computed potential coefficients for all the interactions between the panels in cube  $i$  and its neighbor cubes and pack them into a matrix.*
3. *Invert the constructed potential coefficient matrix.*
4. *Extract the rows corresponding to the panels in cube  $i$  only, and put their entries into the corresponding entries in  $\tilde{C}$ .*
5. *If cube  $i$  is the last non-empty lowest-level cube, stop; otherwise set cube  $i$  to the next non-empty lowest-level cube and return to step 2.*

The preconditioner constructed using Algorithm 3.4 takes into account all interactions between panels that are within a neighbor-cube distance of each other. A preconditioner that includes only those interactions within cube  $i$  when constructing the submatrices in step 2 misses interactions between nearby panels in adjoining cubes. Because it overlooks certain nearby-panel interactions, such a block-diagonal preconditioner does not cluster  $P$ 's eigenvalues as effectively as the preconditioner constructed by Algorithm 3.4.

The three dimensionality of the general problem makes it impossible to number the panels so that all the submatrices extracted form contiguous bands in  $P$  as was possible in the one-dimensional case. Figure 3-14 shows the sparsity pattern formed by the parts of  $P$  used to

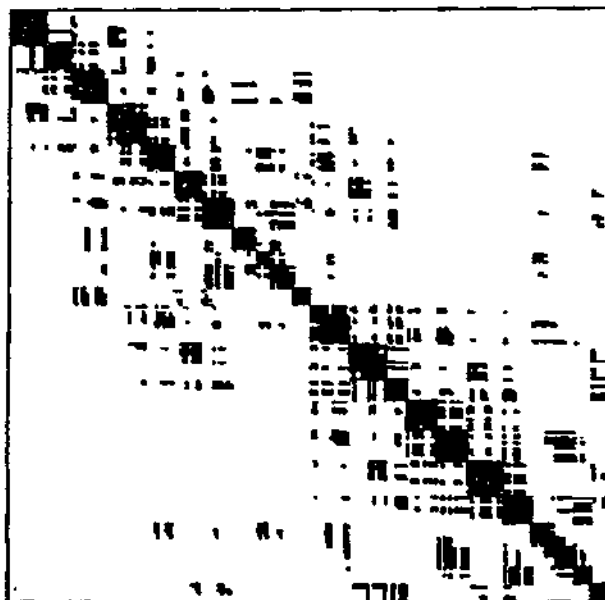


FIGURE 3-14: Dark areas indicate the position of those entries of  $P$  used to form the preconditioner for the woven1 discretization of Figure 4-9.

form  $\tilde{C}$  for a typical three-dimensional problem. These parts of  $P$  are exactly those potential coefficients calculated for the nearby interactions of the multipole algorithm. The resulting preconditioner has the same sparsity as the extracted parts of  $P$ , precluding the use of the generalized Schur algorithm [95, 134, 38]. The generalized Schur algorithm exploits  $P$ 's band structure for two-dimensional problems to find a positive-definite approximation of  $P$ 's inverse. In contrast, the preconditioner used here is not guaranteed to be definite or normal, but is applicable to three-dimensional problems with random nearby-interaction sparsity like that pictured in Figure 3-14.



### 3.3 Extending the Algorithm to Problems with Multiple Dielectrics

The multipole-accelerated capacitance calculation algorithm described in Sections 3.1 and 3.2 applies only to problems with conductors in regions filled with a homogeneous dielectric. This section describes how problems with arbitrary dielectrics can be formulated as equivalent free-space problems, thereby allowing the application of the multipole algorithm. The development of a multipole-accelerated capacitance calculation method for problems with multiple isotropic dielectric regions, each with a different permittivity follows the same path as the homogeneous-dielectric algorithm developed in previous sections.

#### 3.3.1 Changing the Mathematical Formulation

Two methods are useful for including piecewise constant dielectric regions in integral-equation formulations of the capacitance problem.

##### Green's Function Techniques

A capacitance problem involving any system of dielectrics can in principle be solved using the Green's function technique of Section 2.1.1. Besides the difficulty of finding the Green's function for the conductor and dielectric geometry, the method is hampered by the extra Gauss's Law step. Alternatively the conductors can be converted to charge shells as in the free-space case (Section 2.1.2) and related to the conductor potentials by the superposition integral

$$f(x) = \iint_{S_c} \sigma(x') G_d(x, x') dS'_c, \quad x \in S_c.$$

Here  $S_c$  is the conductor surfaces and  $f$  is as in (2.2). The Green's function  $G_d(x, x')$  corresponds to the original problem with the conductors removed, making it simpler than its counterpart in the formulation of Section 2.1.1. The dielectric Green's function approach is particularly useful when the dielectrics have a special geometry, for example stratified media [99, 35, 1]. Techniques of this sort can also be used for anisotropic media [67, 12]. Except for simple cases, however, the complexity of the Green's function makes multipole algorithm acceleration difficult.

##### Charge Layer Approach

The approach taken here allows more general geometries but applies only to linear isotropic dielectrics [63, 105]. With isotropic dielectrics, the exterior Dirichlet problem is identical to (2.1–2.3) with additional boundary conditions at the dielectric interfaces,

$$\nabla^2 \psi(x) = 0, \quad x \in V; \tag{3.32}$$

$$\psi(x) = f(x), \quad x \in S_c; \tag{3.33}$$

$$\lim_{x \rightarrow \infty} \psi(x) = 0; \quad (3.34)$$

$$\epsilon_b \frac{\partial \psi_b(x)}{\partial n_b} = \epsilon_a \frac{\partial \psi_a(x)}{\partial n_b}, \quad x \in S^{ab}, \quad \text{for all } a, b, \quad (3.35)$$

with  $a$  and  $b$  corresponding to bordering dielectrics. Here  $S^{ab}$  is the interface surface between dielectrics  $a$  and  $b$  and  $n_b$  is the normal to  $S^{ab}$  at  $x$  that points into dielectric  $b$ .

The new boundary conditions (3.35) are added to the Section 2.1.2 formulation using a generalized version of the Section 2.1.2 approach. The dielectric boundaries as well as the conductor-dielectric interfaces are replaced by a surface-charge density  $\sigma$  and the dielectrics and conductors are replaced by free space. The conductor potentials are fixed using the integral equation (2.9)

$$f(x) = \iint_S \sigma(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|_2} dS', \quad x \in S_c. \quad (3.36)$$

Here  $S$  is the union of all the dielectric interfaces  $S^{ab}$  and the conductor surfaces  $S_c$ . Any solution to (3.36) satisfies (3.32–3.34). The remaining equations come from the dielectric boundary conditions (3.35). The normal derivatives of the surface potentials on  $S^{ab}$  are (see Appendix A)

$$\frac{\partial \psi_b(x)}{\partial n_b} = -\frac{1}{2\epsilon_0} \sigma(x) - \iint_S \sigma(x') \frac{\cos(x - x', n_b)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS', \quad x \in S^{ab}; \quad (3.37)$$

$$\frac{\partial \psi_a(x)}{\partial n_b} = \frac{1}{2\epsilon_0} \sigma(x) - \iint_S \sigma(x') \frac{\cos(x - x', n_b)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS', \quad x \in S^{ab}, \quad (3.38)$$

using the notation of (3.35). Here  $\psi_a$  and  $\psi_b$  are the limiting free-space potentials in the former dielectric regions. In order for the free-space potentials to be a solution to (3.32–3.35), the normal derivative boundary conditions with (3.37) and (3.38) substituted,

$$0 = \frac{(\epsilon_a + \epsilon_b)}{2\epsilon_0} \sigma(x) - (\epsilon_a - \epsilon_b) \iint_S \sigma(x') \frac{\cos(x - x', n_b)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS', \quad x \in S^{ab}, \quad (3.39)$$

must hold for all dielectric interface surfaces  $S^{ab}$ . The simultaneous solution of (3.36) and (3.39) for  $\sigma$  gives capacitance matrix entries by integration over the conductor surfaces.

### 3.3.2 Weighted-Residuals Discretization

Applying the method of weighted residuals exactly as in Section 2.2.1 converts the equations to a linear system relating the surface-charge density expansion weights to the conductor potentials. The discretization starts with a finite-element expansion of the unknown surface-charge density  $\sigma$ ,

$$\sigma(x) \approx \sum_{j=1}^n a_j w_j(x), \quad x \in S, \quad (3.40)$$

where the  $n$  expansion functions,  $w_j$ , are non-zero only on triangular or quadrilateral regions that panel the conductor and dielectric interface surfaces,

$$w_j(x) \triangleq \begin{cases} 1 & x \in \text{panel } j; \\ 0 & \text{otherwise.} \end{cases}$$

Substituting the expansion (3.40) into (3.36) and (3.39) gives two residuals,

$$R_1(x) \triangleq f(x) - \iint_S \sum_{j=1}^n a_j w_j(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|_2} dS', \quad x \in S_c; \quad (3.41)$$

$$R_2(x) \triangleq \frac{(\epsilon_a + \epsilon_b)}{2\epsilon_0} \sum_{j=1}^n a_j w_j(x) - (\epsilon_a - \epsilon_b) \iint_S \sum_{j=1}^n a_j w_j(x') \frac{\cos(x - x', n_b)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS', \quad x \in S^{ab}; \quad (3.42)$$

which are simultaneously zero when the  $a_j$ 's are such that (3.40) solves (3.36) and (3.39) exactly. Assuming the problem is well conditioned, finding expansion weights  $a_j$  that make both residuals approximately zero gives a good solution to (3.36) and (3.39). The nature of the approximation depends on the testing method used.

### Collocation Testing

Collocation produces a linear system for the unknown  $a_j$ 's by forcing (3.41) and (3.42) to hold at the centroid of the conductor and dielectric-interface panels respectively. If the  $i$ -th panel is a conductor panel with centroid  $x_i$ , then the collocation method applied to (3.41) at  $x_i$  gives

$$0 = f(x_i) - \iint_S \sum_{j=1}^n a_j w_j(x') \frac{1}{4\pi\epsilon_0 \|x_i - x'\|_2} dS', \quad x_i \in S_c. \quad (3.43)$$

When the  $i$ -th panel is a dielectric interface panel on  $S^{ab}$  then (3.42) leads to

$$0 = \frac{(\epsilon_a + \epsilon_b)}{2\epsilon_0} \sum_{j=1}^n a_j w_j(x_i) - (\epsilon_a - \epsilon_b) \iint_S \sum_{j=1}^n a_j w_j(x') \frac{\cos(x_i - x', n_b)}{4\pi\epsilon_0 \|x_i - x'\|_2^2} dS', \quad x_i \in S^{ab}. \quad (3.44)$$

If there are  $n_c$  conductor panels and  $n_d$  dielectric-interface panels then  $n = n_c + n_d$  and  $n_c$  equations of the form (3.43) combined with  $n_d$  equations of the form (3.44) form an  $n \times n$  linear system for the unknown weights  $a_j$ . The weight  $a_j$  has the physical interpretation of the constant charge density on panel  $j$ ,

$$a_j = \frac{q_j}{A_j}, \quad (3.45)$$

where  $q_j$  and  $A_j$  are panel  $j$ 's total charge and area respectively. Substituting (3.45) transforms (3.43) and (3.44) into prototypical rows in a linear system for the  $q_j$ 's. Since  $w_j$  is non-zero only on the  $j$ -th panel, (3.43) and (3.44) become

$$f(x_i) = \sum_{j=1}^n q_j \frac{1}{A_j} \iint_{\text{panel } j} \frac{1}{4\pi\epsilon_0 \|x_i - x'\|_2} dS', \quad x_i \in S_c; \quad (3.46)$$

$$0 = q_i \frac{(\epsilon_a + \epsilon_b)}{2A_i\epsilon_0(\epsilon_a - \epsilon_b)} - \sum_{j=1}^n q_j \frac{1}{A_j} \iint_{\text{panel } j} \frac{\cos(x_i - x', n_b)}{4\pi\epsilon_0 \|x_i - x'\|_2^2} dS', \quad x_i \in S^{ab}, \quad (3.47)$$

where the dielectric rows have been divided by  $(\epsilon_b - \epsilon_a)$  so that dielectric information is only required for the  $q_i$  term.

The linear system represented by (3.46) and (3.47) can be written in matrix form

$$\begin{bmatrix} P \\ E \end{bmatrix} [q] = \begin{bmatrix} f \\ 0 \end{bmatrix}, \quad (3.48)$$

where  $P$  is an  $n_c \times n$  potential coefficient matrix with entries given by (3.4) and  $E$  is an  $n_d \times n$  matrix of normal-voltage-derivative, or normal-electric-field coefficients,

$$E_{ij} \triangleq \begin{cases} \frac{1}{A_j} \iint_{\text{panel } j} \frac{\cos(x_i - x', n_b)}{4\pi\epsilon_0 \|x_i - x'\|_2^2} dS', & i \neq j; \\ \frac{(\epsilon_a + \epsilon_b)}{2A_i\epsilon_0(\epsilon_b - \epsilon_a)}, & i = j, \end{cases} \quad (3.49)$$

which may be calculated using closed form expressions [105]. The special case  $i = j$  results because  $\cos(x_j - x', n_b) = 0$  when  $x' \in \text{panel } j$ . The unknown vector  $q$  is the vector of the  $n$  panel charges, and the right-hand-side vector is formed by combining  $f$ , formed from the  $n_c$   $f(x_i)$  values, with a vector of  $n_d$  zeros. The definitions

$$A \triangleq \begin{bmatrix} P \\ E \end{bmatrix}; \quad b \triangleq \begin{bmatrix} f \\ 0 \end{bmatrix} \quad (3.50)$$

allow the more compact equation

$$Aq = b. \quad (3.51)$$

### Collocation-Galerkin Testing

The collocation discretization, while adequate for problems without dielectric interfaces, is prone to errors for certain geometries with large permittivity changes (see Section 4.4.1). For those problems it is sufficient to use higher accuracy Galerkin testing for (3.42), while still using computationally cheaper collocation for (3.41), as in the previous section.

Applying the Galerkin method to (3.42) forms inner products between  $R_2(x)$  and the  $n_d$  expansion functions  $w_i$  corresponding to panels on the dielectric interfaces,

$$0 = \frac{(\epsilon_a + \epsilon_b)}{2\epsilon_0} \iint_S w_i(x) \sum_{j=1}^n a_j w_j(x) dS \\ - (\epsilon_a - \epsilon_b) \iint_S w_i(x) \iint_S \sum_{j=1}^n a_j w_j(x') \frac{\cos(x - x', n_b)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS' dS, \quad x \in S^{ab}.$$

Since  $w_i$  is non-zero only on panel  $i$ , and the expansion weights can be written in terms of the total panel charges using (3.45), the dielectric-panel row becomes

$$0 = q_i \frac{(\epsilon_a + \epsilon_b)}{2\epsilon_0} \frac{1}{A_i} \iint_{\text{panel } i} dS \\ - (\epsilon_a - \epsilon_b) \sum_{j=1}^n a_j \iint_{\text{panel } i} \iint_{\text{panel } j} \frac{\cos(x - x', n_b)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS' dS, \quad x \in S^{ab};$$

$$0 = q_i \frac{(\epsilon_a + \epsilon_b)}{2A_i \epsilon_0 (\epsilon_b - \epsilon_a)} + \sum_{j=1}^n q_j \frac{1}{A_i A_j} \iint_{\text{panel } i} \iint_{\text{panel } j} \frac{\cos(x - x', n_b)}{4\pi \epsilon_0 \|x - x'\|_2^2} dS' dS, \quad x \in S^{ab}, \quad (3.52)$$

where the final equation has been divided by  $(\epsilon_b - \epsilon_a)A_i$ .

Combining (3.46) with (3.52) gives the hybrid collocation-Galerkin discretization,

$$\begin{bmatrix} P \\ E^G \end{bmatrix} [q] = \begin{bmatrix} f \\ 0 \end{bmatrix}, \quad (3.53)$$

where  $P$  is the  $n_c \times n$  potential coefficient matrix in (3.48) and  $E^G$  is an  $n_d \times n$  matrix of average normal-voltage-derivative coefficients,

$$E_{ij}^G \triangleq \begin{cases} \frac{1}{A_i A_j} \iint_{\text{panel } i} \iint_{\text{panel } j} \frac{\cos(x - x', n_b)}{4\pi \epsilon_0 \|x - x'\|_2^2} dS' dS, & i \neq j; \\ \frac{(\epsilon_a + \epsilon_b)}{2A_i \epsilon_0 (\epsilon_b - \epsilon_a)}, & i = j. \end{cases} \quad (3.54)$$

Since closed form expressions for  $E_{ij}^G$ ,  $i \neq j$ , are known only for special cases [109], numerical approximations like those described in Appendix C are necessary. The definition

$$A^H \triangleq \begin{bmatrix} P \\ E^G \end{bmatrix},$$

together with (3.50) allow the more compact equation

$$A^H q = b. \quad (3.55)$$

### 3.3.3 Multipole Accelerated Iterative Solution

In order to avoid the  $O(n^3)$  time of Gaussian elimination, the linear system (3.51) or (3.55) can be solved using an iterative method. The reasoning of Section 3.2.1 leads to GMRES as the best choice, giving  $O(n^2)$  time and storage. Application of the multipole algorithm further reduces the time to nearly  $O(mn)$ , where  $m$  is the number of conductors, and the storage to nearly  $O(n)$ . The multipole algorithm applies directly to the  $Pq$  products in (3.50) and (3.53) in exactly the same way as described in Section 3.2.2, while the  $Eq$  and  $E^G q$  products require different use of the multipole algorithm. Extending the preconditioner used in the homogeneous dielectric case (see Section 3.2.4) keeps the number of GMRES iterations reasonable for arbitrary geometries.

#### Accelerated Potential Evaluations

The GMRES iterative method has the form of Algorithm 3.1, with the matrix-vector product dominating the cost of one iteration. When applied to (3.51) or (3.55), the part of the

matrix-vector product corresponding to the conductor panel equations,  $Pq$ , is equivalent to calculating the potential at the centroids of all  $n_c$  conductor panels due to charge on all  $n$  panels. This calculation can be done more efficiently than the explicit matrix-vector product using the adaptive multipole algorithm, Algorithm 3.3. As described in Section 3.2.2, the multipole method approximates the part of the potential due to the panels distant from each panel centroid, leading to time and memory savings.

### Accelerated Potential Normal Derivative Evaluations

When GMRES is applied to (3.51) or (3.55), the part of the matrix-vector product corresponding to the dielectric-panel equations is  $Eq$  or  $E^Gq$  respectively. If  $i \neq j$ ,  $E_{ij}$  is the contribution of the charge on panel  $j$  to the normal derivative of the potential at panel  $i$ 's centroid, while  $E_{ij}^G$  is the contribution of the charge on panel  $j$  to the average normal derivative of the potential on panel  $i$ . The standard multipole algorithm calculates only potentials, but it can be either generalized to approximate these normal derivative quantities directly, or combined with a post-processing step to do the approximation. The generalized approach incorporates the differentiation, and the subsequent integration in the case of  $E^Gq$ , into the evaluation transformations. The result is an extension of the adaptive multipole algorithm (Algorithm 3.5 below) that is capable of the normal-derivative and average-normal-derivative evaluations in addition to the usual potential evaluations. The other alternative is to use a divided-difference constructed using adaptive-multipole-algorithm potential evaluations to approximate the normal derivatives (Algorithm 3.6 below). This method requires more evaluations than the generalized multipole algorithm technique, but avoids modification of the multipole algorithm.

In general any potential evaluation done with the adaptive multipole algorithm (Algorithm 3.3) uses multipole-expansion evaluations (M2P), local-expansion evaluations (L2P), and direct Coulombic evaluations (Q2P). Evaluation of the normal derivative of the potential is possible by analytically differentiating all the expansions and potential coefficients involved in the potential evaluation. Since differentiation is linear, it is possible to combine the differentiation with the evaluation step, giving a multipole-to-potential-normal-derivative, or M2E, transformation, and a local-to-potential-normal-derivative, or L2E, transformation. Appendix B describes the L2E and M2E operators in more detail. The direct evaluations use normal derivatives of the  $P_{ij}$ 's, namely the  $E_{ij}$ 's, to compute normal derivatives of the potential. That operation is summarized by submatrices of  $E$  called Q2E transformation matrices. Since both the potential (M2P, L2P, Q2P) and normal-derivative (M2E, L2E, Q2E) evaluations use the same multipole and local expansions, and panel charges as in Algorithm 3.3, this collocation-based generalized adaptive multipole algorithm differs from Algorithm 3.3 primarily in the type of transformation matrices used in the evaluation step.

Iterations involving (3.55) must evaluate  $E^Gq$  products which are averages of the normal derivative of the potential over the dielectric panels. Since the averaging process is linear, it can

be combined with the differentiation and evaluation steps to form new linear transformations  $M2E^G$ ,  $L2E^G$  and  $Q2E^G$ , analogous to the corresponding transformations of the collocation-based generalized adaptive multipole algorithm. The resulting collocation-Galerkin generalized adaptive multipole algorithm is generally more accurate than the collocation-based method, but requires more work in the transformation initialization phase.

Both the collocation and collocation-Galerkin approaches lead to a generalized adaptive fast multipole algorithm that efficiently computes the matrix-vector products in a GMRES iterative loop.

**Algorithm 3.5 (Generalized Adaptive Fast Multipole Algorithm)** *To calculate an approximate  $Aq$  product or  $A^H q$  product,*

1. *If this is the first matrix-vector product for a new problem, choose an expansion order  $l$  and compute appropriate transformation matrices for all cubes, taking exact cubes into account, and store for subsequent calls. Included here are  $Q2P$  and  $Q2E$  or  $Q2E^G$  matrices representing self and neighbor evaluations. If matrices are already calculated, go to the next step.*
2. *Perform the upward pass. Apply  $Q2M$  and  $M2M$  matrices to obtain multipole expansions in all non-exact cubes.*
3. *Perform the downward pass. Apply  $M2L$ ,  $L2L$  and  $Q2L$  matrices to obtain local expansions in all non-exact cubes.*
4. *Perform the evaluation pass. Apply  $L2P$ ,  $M2P$  and  $Q2P$  matrices to obtain all collocation point potentials. Apply  $L2E$ ,  $M2E$  and  $Q2E$  matrices, or  $L2E^G$ ,  $M2E^G$  and  $Q2E^G$  matrices, to calculate all normal electric fields or normal field averages respectively. Included here are products with the  $Q2P$  and  $Q2E$  or  $Q2E^G$  matrices representing self and neighbor evaluations.*

A simpler, but slightly more expensive, approach to evaluating the  $Eq$  products using the adaptive multipole algorithm is to approximate the normal derivatives of the potential using divided differences. Rather than modifying the adaptive multipole algorithm (Algorithm 3.3) to give normal electric fields directly as in Algorithm 3.5, the alternative method approximates the normal fields using divided differences constructed from multipole-algorithm potential evaluations. The  $i$ -th row in  $E$  has a diagonal entry representing the contribution of the charge on panel  $i$  to the normal electric-field difference at panel  $i$ , and several off-diagonal entries which incorporate the effects of all other panel charges. Thus the total normal field difference at panel  $i$ 's centroid may be written

$$E_{n_i} = E_{o_i} + E_{ii}q_i, \quad (3.56)$$

where  $E_{o_i}$  is the normal field due to all panels other than panel  $i$ . The field  $E_{o_i}$  can be

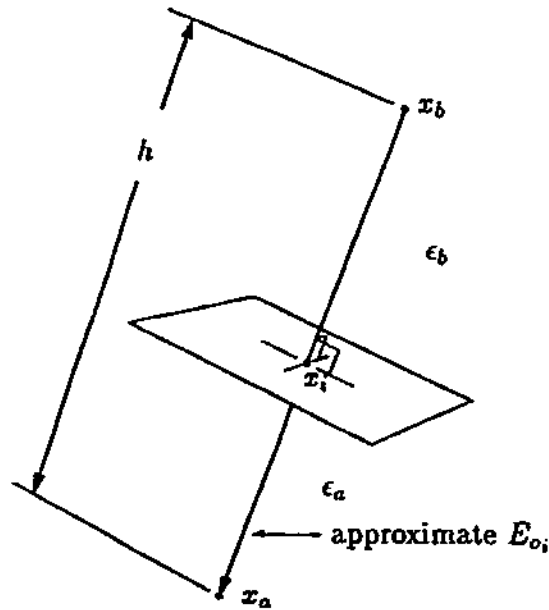


FIGURE 3-15: The part of the normal electric field on the dielectric panel due to distant charge panels can be approximated with a divided difference.

approximated by a divided difference constructed from two potential evaluations as illustrated in Figure 3-15. The potentials at two points  $x_a$  and  $x_b$ , both the same perpendicular distance,  $h/2$ , away from the panel center  $x_i$ , combine to give

$$E_{o_i} \approx \frac{\psi(x_a) - \psi(x_b)}{h}. \quad (3.57)$$

Since the potentials  $\psi(x_a)$  and  $\psi(x_b)$  can be computed using the efficient adaptive multipole algorithm, each row of the  $Eq$  product requires two potential evaluations using the multipole algorithm and one multiply-add.

The divided-difference method also applies to  $E^Gq$  products, but is tied to the approximations of the averaging integral discussed in Appendix C. Those approximations are generalizations of the panel centroid-collocation method used for the  $Eq$  product, which may be viewed as a first-order approximation to the averaging integral over the dielectric-interface panel. Higher-order approximations to the averaging integral combine many normal-field evaluations on each panel in weighted sums. Using a separate divided difference constructed as in Figure 3-15 for each normal-field evaluation on a particular dielectric panel allows the application of the adaptive multipole algorithm to the  $E^Gq$  calculation without modification. A weighted sum of such divided differences, rather than a single divided difference as in the collocation case, replaces the off-diagonal part of the average normal field calculation. Each dielectric-panel average normal field calculation requires several potential evaluations for the divided differences, and one multiply-add for the diagonal term. The extra evaluations generally lead to a more accurate solution than in the centroid-collocation case. The disadvantage of the resulting hybrid divided-



difference method is the need for multiple potential evaluations to compute the many divided differences required for each dielectric-interface panel.

The divided-difference methods for computing  $Aq$  and  $A^Hq$  all have the same form, the only difference being the order of the averaging integral approximation.

**Algorithm 3.6 (Divided-Difference Matrix-Vector Product Calculation)** *To calculate an approximate  $Aq$  or  $A^Hq$ ,*

1. *If this is the first matrix-vector product, choose the order of the normal electric field averaging integral, and then look up the corresponding integration-rule weights and evaluation point positions from Appendix C.*
2. *Use the adaptive multipole algorithm, Algorithm 3.3, to calculate all the conductor-panel collocation point potentials as well as two divided-difference potentials for each dielectric-interface panel evaluation point prescribed by the integration rule.*
3. *Calculate a divided difference and add the diagonal contribution ( $E_{ii}q_i$  in (3.56)) to form a normal electric field difference at each dielectric-panel evaluation point.*
4. *Using the integration rule weights, combine the divided-difference approximations for each dielectric interface panel to obtain approximate average normal-field integrals for all the dielectric panels.*

### 3.3.4 Preconditioning

As in the uniform dielectric case (Section 3.2.4), an effective preconditioner for the multiple dielectric case that captures the stronger, nearby panel interactions can be constructed using only the direct interaction information required by the multipole algorithm. Using essentially the same strategy as for the uniform dielectric problem, the preconditioner rows are extracted from inverted submatrices of  $A$  or  $A^H$  corresponding to interactions between panels in neighboring cubes. The only difference is that the interactions now include normal electric field contributions of all panels to the dielectric panels in addition to the influences of all panels on the potential at conductor panels.

When the iterative loop uses the generalized multipole algorithm (Algorithm 3.5) to compute matrix-vector products, the submatrices of  $A$  or  $A^H$  corresponding to panels in neighboring cubes exist explicitly as  $P_{ij}$  and  $E_{ij}$  coefficients that are entries in Q2P and Q2E or Q2E<sup>G</sup> transformation matrices<sup>11</sup>. In that case the preconditioner calculation proceeds as in the single-dielectric case, except that normal electric-field interactions ( $E_{ij}$ 's) may be present.

---

<sup>11</sup>Since a particular cube can contain both dielectric-interface and conductor panels, these matrices usually have both Q2P and Q2E or Q2E<sup>G</sup> rows, rather than being purely of one type.

**Algorithm 3.7 (Multipole-Accelerated Generalized Preconditioner Build)** *To construct a preconditioner  $\tilde{C}$  for a  $n$ -panel problem using all the entries of  $A$  or  $A^H$  calculated by the generalized multipole algorithm (Algorithm 3.5),*

1. Set cube <sub>$i$</sub>  equal to the first non-empty lowest-level cube and  $\tilde{C} = 0$ .
2. Look up the previously computed potential and normal electric field coefficients for all the interactions between the panels in cube  $i$  and its neighbor cubes and pack them into a matrix.
3. Invert the constructed matrix.
4. Extract the rows corresponding to the panels in cube  $i$  only, and put their entries into the corresponding entries in  $\tilde{C}$ .
5. If cube  $i$  is the last non-empty lowest-level cube, stop; otherwise set cube  $i$  to the next non-empty lowest-level cube and return to step 2.

If the iterative loop uses the divided-difference algorithm (Algorithm 3.6) instead, then only the potential coefficients  $P_{ij}$  are explicitly calculated for use by the unmodified multipole algorithm. The rows of the  $A$  or  $A^H$  submatrices corresponding to dielectric panels must be approximated using the potential coefficients. Consider constructing the submatrix of  $A$  corresponding to the seven-panel example of Figure 3-10, assuming that panel 4 is a dielectric interface panel, and the others are conductor panels. Using the terminology of Figure 3-15, two multipole-algorithm potential evaluations near panel 4 give  $\psi(x_a)$  and  $\psi(x_b)$ , which are influenced by the charge on all the panels,

$$\begin{aligned}\psi(x_a) &= P_{a1}q_1 + P_{a2}q_2 + P_{a3}q_3 + P_{a4}q_4 + P_{a5}q_5 + P_{a6}q_6 + P_{a7}q_7; \\ \psi(x_b) &= P_{b1}q_1 + P_{b2}q_2 + P_{b3}q_3 + P_{b4}q_4 + P_{b5}q_5 + P_{b6}q_6 + P_{b7}q_7.\end{aligned}$$

Using these equations and (3.56) with (3.57) substituted gives the normal electric field difference,

$$E_{n4} = \sum_{\substack{i=1 \\ i \neq 4}}^7 \hat{E}_{4i}q_i + E_{44}q_4, \quad (3.58)$$

where

$$\hat{E}_{4i} \triangleq \frac{P_{ai} - P_{bi}}{h}. \quad (3.59)$$

Here  $h$  is as in Figure 3-15 and  $E_{44}$  is given by (3.49). Panel 4's row approximation to the dielectric panel row,

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{E}_{41} & \hat{E}_{42} & \hat{E}_{43} & E_{44} & \hat{E}_{45} & \hat{E}_{46} & \hat{E}_{47} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (3.60)$$

is implicit in (3.58) and forms one row of the submatrix. The remaining rows are for conductor panels 1-3 and 5-7 and consist entirely of potential coefficients.

The submatrices of  $A^H$  also require approximation for rows corresponding to dielectric panels. As in the  $A$ , or collocation, case, the normal-field evaluations are calculated by divided differences, giving an expression for each evaluation in terms of two potential coefficients. Each dielectric panel has many normal-field evaluation point expressions that are combined in weighted sums as prescribed by the integration rules of Appendix C, giving an approximate dielectric-panel row with the form of (3.60),

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{E}_{41}^G & \hat{E}_{42}^G & \hat{E}_{43}^G & E_{44} & \hat{E}_{45}^G & \hat{E}_{46}^G & \hat{E}_{47}^G \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

for the Figure 3-10 geometry. In this case the normal electric-field approximations for the Figure 3-10 problem have the general form

$$\hat{E}_{4i}^G \triangleq \sum_{j=1}^k d_j \frac{P_{a_j,i} - P_{b_j,i}}{h}, \quad (3.61)$$

where the integration rule has weights  $d_j$  and the  $j$ -th normal electric field evaluation point on the panel corresponds to two off-panel potential evaluation points,  $a_j$  and  $b_j$ . Centroid collocation is the special case resulting when  $a_1$  and  $b_1$  are points near the panel centroid and  $d_1 = k = 1$ .

Combining these ideas gives a preconditioner construction algorithm appropriate for use with iterative loops using Algorithm 3.6 to compute matrix-vector products. The algorithm is the same as Algorithm 3.7 except that it approximates the dielectric-panel rows used to form the preconditioner. Using divided differences, the electric field coefficients are approximated with potential coefficients corresponding to off-panel potential evaluations.

**Algorithm 3.8 (Multipole-Accelerated Divided-Difference Preconditioner Build)** *To construct a preconditioner  $\tilde{C}$  for a  $n$ -panel problem using all the entries of  $A$  or  $A^H$  calculated by the adaptive multipole algorithm (Algorithm 3.9) as a part of the matrix-vector multiplication algorithm, Algorithm 3.6,*

1. Set cube  $i$  equal to the first non-empty lowest-level cube and  $\tilde{C} = 0$ .
2. For each conductor panel in cube  $i$ , look up the previously computed potential coefficients representing the effect of the panel charge in cube  $i$  and its neighbors on the selected panel's centroid potential and store them in a matrix. For each dielectric-interface panel in cube  $i$ , look up the previously computed potential coefficients representing the effect of the panel charge in cube  $i$  and its neighbors on the off-panel potential evaluations specified by the same integration rule chosen for Algorithm 3.6. Combine the off-panel potential

coefficients using (3.59) or (3.61) to approximate the off-diagonal entries of the selected dielectric-interface panel's row. Store these entries, together with the known  $E_{ii}$  in the same matrix with the conductor panel rows.

3. Invert the constructed matrix.
4. Extract the rows corresponding to the panels in cube  $i$  only, and put their entries into the corresponding entries in  $\tilde{C}$ .
5. If cube  $i$  is the last non-empty lowest-level cube, stop; otherwise set cube  $i$  to the next non-empty lowest-level cube and return to step 2.

### 3.3.5 Capacitance Matrix Calculation from the Charge Layer Density

Solving for the vector of panel charges  $q$  gives the approximation (3.1) to the charge density  $\sigma$  which is integrated over the conductor surfaces to calculate capacitance matrix entries. If  $x$  is a point on the boundary between a conductor and dielectric region  $b$  in the equivalent free-space problem, then

$$\sigma(x) = \epsilon_0 \frac{\partial \psi_b(x)}{\partial n_b}. \quad (3.62)$$

The density  $\sigma$  is chosen so that  $\psi_b(x)$  is the same in the corresponding problem with  $\epsilon_b$  dielectric in region  $b$ . Thus the analogous relation for the dielectric problem uses the same potential, but a different charge density,

$$\sigma_f(x) = \epsilon_b \frac{\partial \psi_b(x)}{\partial n_b}. \quad (3.63)$$

The surface-charge density  $\sigma_f(x)$  is called the free-charge density. It is the actual surface-charge density induced on the conductor, the density that is integrated to give capacitance matrix entries. Combining (3.62) and (3.63) gives

$$\sigma_f(x) = \frac{\epsilon_b}{\epsilon_0} \sigma(x), \quad (3.64)$$

so that  $\sigma_f(x)$  is proportional to the  $\sigma(x)$  obtained by solving (3.36) and (3.39) simultaneously.

In the case of an infinitesimally thin conductor lying on the interface between  $\epsilon_b$  and  $\epsilon_a$  dielectrics, the equation analogous to (3.63) is

$$\sigma_f(x) = \epsilon_a \frac{\partial \psi_a(x)}{\partial n_b} - \epsilon_b \frac{\partial \psi_b(x)}{\partial n_b}.$$

Substituting (3.37) and (3.38) gives an integral expression for  $\sigma_f(x)$  in terms of  $\sigma(x)$ ,

$$\sigma_f(x) = \frac{\epsilon_a + \epsilon_b}{2\epsilon_0} \sigma(x) - (\epsilon_a - \epsilon_b) \iint_S \sigma(x') \frac{\cos(x - x', n_b)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS', \quad x \text{ on thin conductor.}$$

This expression is substituted for (3.64) when the conductor region is infinitesimally thin.

---

---

# Multipole Accelerated Capacitance Calculation Algorithm Evaluation

Combining the techniques of Chapter 3 to implement a capacitance calculation algorithm provides a concrete foundation for evaluating the new method. Section 4.1 describes the particular implementation used here, and later sections discuss the properties of the complete algorithm, using the implementation as a guide. Sections 4.2 and 4.3 discuss the algorithm's convergence and error properties and computational complexity, while Section 4.4 demonstrates its performance in actual capacitance calculations.

## 4.1 Implementation

This section gives a detailed description of an implementation of a capacitance-calculation algorithm based on a preconditioned, adaptive multipole accelerated GMRES iterative method. The iterative loop has the form of Algorithm 3.1 with iterates calculated using Algorithm 3.6, and uses a preconditioner calculated with Algorithm 3.8. Section 4.1.1 describes the main capacitance calculation routine, while Sections 4.1.2 and 4.1.3 discuss the multipole algorithm initialization and field-calculation functions respectively<sup>1</sup>.

### 4.1.1 Capacitance Calculation Algorithm

The capacitance calculation algorithm implemented here uses the weighted-residuals discretization of the integral-equation formulation described in Section 3.3.2. The method uses panel-centroid collocation on the conductor panels and Galerkin testing on dielectric-interface panels, following Section 3.3.2. Several divided-difference approximations to the normal electric field on each dielectric-interface panel approximate the Galerkin integrals using the integration

---

<sup>1</sup>The actual C program implementation of the algorithm, *fastcap*, is available by anonymous ftp from [rle-vlsi.mit.edu](http://rle-vlsi.mit.edu).

rules of Appendix C, with panel-centroid collocation a first-order special case. The resulting linear system is solved with preconditioned GMRES, using the divided-difference approach to calculate the matrix-vector products (Algorithm 3.6), and the preconditioner (Algorithm 3.8).

The implementation uses standard GMRES [112] to solve (3.55),

$$A^H q = b,$$

for the unknown vector of panel charges  $q$ , with  $m$  different  $b$ 's given by the potential problems associated with the  $m$  conductors. The function `Input_Problem(input_files)` reads the problem description from the files listed in `input_files`, producing a list of conductors, `conductor_list`, a list of all the conductor and dielectric-interface panels, `panel_list`, the spatial-partitioning depth, `depth`, the dielectric-panel Galerkin-integral approximation order,  $o$ , the expansion order,  $l$ , and the GMRES iterative-loop tolerance, `GMRES $tol$` . The default value of `depth` is `auto`, which causes the partitioning depth to be chosen automatically based on the problem geometry, while  $o = 1$  (centroid collocation),  $l = 2$  and `GMRES $tol$`  = 0.01 by default. After reading the problem description, the adaptive multipole algorithm's data structures are initialized using `Initialize_Multipole()`, which returns the panel-charge and evaluation-point potential vectors,  $q$  and  $p$  respectively. Each pass through the main loop calculates the column of the capacitance matrix corresponding to a different conductor,  $j$ , by first solving for  $q$  with GMRES, and then using `Sum_Charge( $q, i$ )` to figure each  $c_{ij}$  by adding the conductor- $i$  panel charges, scaled by the relative permittivities of the bordering dielectrics times  $4\pi\epsilon_0$ . The factor of  $4\pi\epsilon_0$  is absorbed into the definition of charge for the purposes of the calculation, leading to greater efficiency and a more convenient scaling<sup>2</sup>. The final step averages off-diagonal entries of the capacitance matrix to obtain a symmetric matrix.

Since `Load_Initial_Residual( $q, j$ )` loads  $q$  with the right-hand-side vector  $b$  corresponding to conductor  $j$  at unit potential, the others at zero, the GMRES loop begins with the zero vector as the initial guess. The initial call to `Apply_Preconditioner( $q$ )` returns  $\tilde{C}q$ , applying the preconditioner to  $b$ . Most of the work is in the inner, iterative loop in the repeated calls to `Calculate_Multipole_Fields( $p, q$ )`, which implements Algorithm 3.6 followed by multiplication by the Algorithm 3.8 preconditioner. Subsequent sections give further descriptions of these functions, and the multipole initialization routine, `Initialize_Multipole(panel_list, depth,  $l$ )`.

Algorithm 4.1 is a realization of the complete preconditioned adaptive multipole accelerated GMRES iterative method, referred to as PAMA. The same algorithm without the `Apply_Preconditioner()` calls in Algorithms 4.1 and 4.3, and the preconditioner construction step, Phase 5, in Algorithm 4.2 is the adaptive multipole accelerated GMRES iterative algorithm, referred to as AMA. If in addition all the if statements involving comparisons of

---

<sup>2</sup>In other words the calculation uses Gaussian units, but the results are given in MKSA. For a comparison of these units see [61].

vector lengths to  $(l + 1)^2$  in Algorithm 4.2 are made unconditional, then the multipole algorithm builds expansions at all non-empty cubes, resulting in the multipole accelerated GMRES iterative algorithm, or MA for brevity. The term “multipole accelerated algorithm” also refers to all these implementations collectively, as does the shorthand FASTCAP.

#### Algorithm 4.1 (Capacitance Calculation Algorithm)

```

/* Problem Input and Multipole Initialization */
[conductor_list, panel_list, depth, o, l, GMRES tol] = Input_Problem(input_files).
[p, q] = Initialize_Multipole(panel_list, depth, o, l).
/* Compute Capacitances Associated with Each Conductor */
for each conductor j in conductor_list {
    /* Perform GMRES Iterative Loop Solve for q */
    Load_Initial_Residual(q, j).      /* Temporarily store  $\tilde{C}b$  in q */
    Apply_Preconditioner(q).
     $\|\tilde{C}b\|_2 = \|q\|_2$ .
     $r^0 = v^0 = q$ .
    k = 1.
    /* Construct the Krylov Subspace Vectors */
    while ( $\|r^{k-1}\|_2 / \|\tilde{C}b\|_2 > GMRES\ tol$ ) {
        /* Approximate the Preconditioned Matrix-Vector Product */
         $q = v^{k-1}$ .
        Calculate_Multipole_Fields(p, q).
        /* Find Projections of New Krylov Vector on Old Krylov Vectors */
        for each previously computed  $v^i$ ,  $0 \leq i < k$  {
             $h_{ik} = p^T v^i$ .
        }
         $p = p - \sum_{i=0}^{k-1} h_{ik} v_i$ .          /* Orthogonalize */
         $h_{kk} = \|p\|_2$ .                    /* Normalize */
         $v_k = p / h_{kk}$ .
        /* Compute the Residual */
        compute  $\|r^k\|_2$  using the  $h_{ij}$ 's.
        k = k + 1.
    }
    /* Construct the Solution */
    choose  $\alpha_j$ 's in  $q = \sum_{j=0}^{k-1} \alpha_j v^j$  to minimize  $\|r^{k-1}\|$ .
    /* Sum Charges to Get Capacitances */
    for each conductor i in conductor_list {
         $c_{ij} = \text{Sum\_Charge}(q, i)$ .
    }
}

```

```

}
/* Symmetrize the Capacitance Matrix */
for each conductor  $i$  in conductor_list {
  for each conductor  $j > i$  in conductor_list {
     $c_{ij} = c_{ji} = (c_{ij} + c_{ji})/2.$ 
  }
}

```

#### 4.1.2 Multipole Algorithm Initialization

The capacitance calculation algorithm (Algorithm 4.1) calls `Initialize_Multipole()` once to build the data structure that contains all the geometry-dependent information of the problem. The cost of the initialization is amortized over the many calls to `Calculate_Multipole_Fields()` performed in the inner loop of Algorithm 4.1.

The names of functions and variables that are particular to the implementation used for this thesis are set in typewriter font and are discussed after the algorithm description. The remaining symbols represent the cube hierarchy with  $D$  levels (see Section 3.2.2) and associated quantities. A cube on level  $d$  with index  $j$  contains panels carrying total charges given by the vector  $q_{d,j}$ , and evaluation points with potentials given by the vector  $p_{d,j}$ . All cubes devoid of panels or evaluation points are not included in the hierarchy. These vectors are contiguous sub-vectors of the vectors  $q$  and  $p$ , which represent all the panel charges and evaluation point potentials in the problem. The cubes may have a multipole expansion, denoted  $M_{d,j}$ , or a local expansion, denoted  $L_{d,j}$ , with order  $l$ . The algorithm uses Q2M, M2M, M2L, L2L, L2P, Q2P, Q2L and M2P matrices to calculate these vectors as described in Section 3.2.3. Their arguments indicate source and destination cubes. For example,  $Q2M(d, j, \tilde{d}, \tilde{j})$  produces  $M_{d,j}$  from  $q_{\tilde{d},\tilde{j}}$ . All lowest-level cubes in the hierarchy have  $\tilde{C}$  matrices which together store the divided-difference preconditioner, generated using Algorithm 3.8 as described in Section 3.3.4. The remaining quantities describe the cube's position in the hierarchy.

$F(d, j)$  is the parent of cube  $j$  on level  $d$ .  $F$  with no argument refers to the set of all parent cubes.

$C(d, j)$  is the set of non-empty children of cube  $j$  on level  $d$ .  $C$  with no argument refers to the set of all child cubes.

$N(d, j)$  is the set of neighbor cubes for cube  $j$  on level  $d$ . The neighbor cube set is the non-empty nearest neighbors and second-nearest neighbors of the given cube, and the given cube itself. The nearest neighbors are those level  $d$  cubes that share a corner with the given cube; the second-nearest neighbors are the level  $d$  cubes that share a corner with the nearest neighbors, but are not nearest neighbors themselves. In two dimensions, as



illustrated by the examples of previous sections, the neighbor region does not include second-nearest neighbors.  $N$  with no argument refers to the set of all neighbor cube sets.

$I(d, j)$  is the set or list of interaction cubes for cube  $j$  on level  $d$ . The set is defined in terms of the mixed-size interaction set, which is the interaction list of the original algorithm [52, 54], with level  $d - 1$  cubes included when they do not lead to expansion evaluations which violate the constraint (3.9). More succinctly, the mixed-size interaction set contains all non-empty cubes on level  $d - 1$  which are second-nearest neighbors of  $F(d, j)$ , and all non-empty level  $d$  cubes that are children of  $F(d, j)$ 's nearest neighbors and are not in  $N(d, j)$ . For cubes with  $F(d, j) = \text{Adaptive\_Parent\_Cube}(d, j)$ ,  $I(d, j)$  is the mixed-size interaction set. For lowest-level cubes, or cubes that have more than one non-empty child cube and an adaptive parent that is different from their parent,  $I(d, j)$  is the set of all the mixed-size interaction sets for all ancestors of cube  $j$  on level  $d$  that are also descendants of its adaptive parent cube, including the  $j, d$  cube itself. All other cubes have  $I(d, j)$  equal to the empty set.  $I$  with no argument refers to the set of all interaction cube sets.

Only `Initialize_Multipole()` uses the cube hierarchy explicitly to determine the sequence of the many transformation matrix-vector products that make up the calculation, as well as the matrices and their source and destination vectors. The sequence information is stored in lists of source vector, transformation matrix and destination vector triples, with one list for each phase of the calculation. Subsequent application of the multipole algorithm with `Calculate_Multipole_Fields()` requires only performing the matrix-vector products in the order specified by the lists, thereby avoiding repeated cube-relation checks and transformation matrix calculations.

**Algorithm 4.2** (`Initialize_Multipole(panel_list, depth, o, l)`)

*/\* Phase 0: Multipole Data Structure Build and Initialization \*/*

$[F, C, p, q, D] = \text{Build\_Cube\_Hierarchy}(\text{panel\_list}, \text{depth}, o, l).$

$N = \text{Build\_Neighbor\_Sets}(F, C).$

$I = \text{Build\_Interaction\_Sets}(F, C).$

*/\* Phase 1: Direct Multipole Operator Build \*/*

for each finest-level cube  $j = 1$  to  $8^D$  {

for each of cube  $j$ 's neighbor cubes  $\bar{j} \in N(D, j)$  {

$Q2P(D, j, D, \bar{j}) = \text{Calculate\_Potential\_Coefficients}(D, j, D, \bar{j}).$

$\text{Add\_To\_List}(\text{direct\_list}, p_{D,j}, Q2P(D, j, D, \bar{j}), q_{D,j}).$

}

}

*/\* Phase 2: Upward Pass Multipole Operator Build \*/*

```

for each finest-level cube  $j = 1$  to  $8^D$  {
  if (Length( $q_{D,j}$ ) >  $(l+1)^2$ ) {
     $Q2M(D, j, D, j) = \text{Calculate\_Charge\_To\_Multipole}(D, j, D, j)$ .
    Add_To_List(up_list,  $M_{D,j}, Q2M(D, j, D, j), q_{D,j}$ ).
  }
}
for each level  $d = D - 1$  to  $2$  {
  for each level  $d$  cube  $j = 1$  to  $8^d$  {
    if (Length( $q_{d,j}$ ) >  $(l+1)^2$ ) {
      if (Size( $C(d, j)$ ) > 1) {
        for each level  $d+1$  cube  $\bar{j} \in C(d, j)$  {
          if (Length( $q_{d+1,\bar{j}}$ ) >  $(l+1)^2$ ) {
             $M2M(d, j, d+1, \bar{j}) = \text{Calculate\_Multipole\_To\_Multipole}(d, j, d+1, \bar{j})$ .
            Add_To_List(up_list,  $M_{d,j}, M2M(d, j, d+1, \bar{j}), M_{d+1,\bar{j}}$ ).
          }
          else {
             $Q2M(d, j, d+1, \bar{j}) = \text{Calculate\_Charge\_To\_Multipole}(d, j, d+1, \bar{j})$ .
            Add_To_List(up_list,  $M_{d,j}, Q2M(d, j, d+1, \bar{j}), q_{d+1,\bar{j}}$ ).
          }
        }
      }
    }
  }
  else {
    Add_To_List(up_list,  $M_{d,j}, \text{identity}, M_{d+1,j}$ ).
  }
}
}
}
}

```

**/\* Phase 3: Downward Pass Multipole Operator Build \*/**

```

for each level  $d = 2$  to  $D$  {
  for each level  $d$  cube  $j = 1$  to  $8^D$  with Size( $I(d, j)$ ) > 0 {
    [level, cube] = Adaptive_Parent_Cube( $d, j$ ).
    if (Length( $p_{d,j}$ ) >  $(l+1)^2$ ) {
      Set_Local_Expansion_Flag( $d, j$ ).
      if (level > 1) {
         $L2L(d, j, \text{level}, \text{cube}) = \text{Calculate\_Local\_To\_Local}(d, j, \text{level}, \text{cube})$ .
        Add_To_List(down_list,  $L_{d,j}, L2L(d, j, \text{level}, \text{cube}), L_{\text{level}, \text{cube}}$ ).
      }
    }
  }
}

```

```

for each  $\bar{d}, \bar{j} \in I(d, j)$  {
  if (Length( $q_{\bar{d}, \bar{j}}$ ) >  $(l + 1)^2$ ) {
     $M2L(d, j, \bar{d}, \bar{j}) = \text{Calculate\_Multipole\_To\_Local}(d, j, \bar{d}, \bar{j})$ .
    Add_To_List(down_list,  $L_{d, j}, M2L(d, j, \bar{d}, \bar{j}), M_{\bar{d}, \bar{j}}$ ).
  }
  else {
     $Q2L(d, j, \bar{d}, \bar{j}) = \text{Calculate\_Charge\_To\_Local}(d, j, \bar{d}, \bar{j})$ .
    Add_To_List(down_list,  $L_{d, j}, Q2L(d, j, \bar{d}, \bar{j}), q_{\bar{d}, \bar{j}}$ ).
  }
}
}
else {
  if (Local_Expansion_Flag_Set(level, cube) = 1) {
     $L2P(d, j, \text{level}, \text{cube}) = \text{Calculate\_Local\_To\_Potential}(d, j, \text{level}, \text{cube})$ .
    Add_To_List(down_list,  $p_{d, j}, L2P(d, j, \text{level}, \text{cube}), L_{\text{level}, \text{cube}}$ ).
  }
  for each  $\bar{d}, \bar{j} \in I(d, j)$  {
    if (Length( $q_{\bar{d}, \bar{j}}$ ) >  $(l + 1)^2$ ) {
       $M2P(d, j, \bar{d}, \bar{j}) = \text{Calculate\_Multipole\_To\_Potential}(d, j, \bar{d}, \bar{j})$ .
      Add_To_List(down_list,  $p_{d, j}, M2P(d, j, \bar{d}, \bar{j}), M_{\bar{d}, \bar{j}}$ ).
    }
    else {
       $Q2P(d, j, \bar{d}, \bar{j}) = \text{Calculate\_Potential\_Coefficients}(d, j, \bar{d}, \bar{j})$ .
      Add_To_List(down_list,  $p_{d, j}, Q2P(d, j, \bar{d}, \bar{j}), q_{\bar{d}, \bar{j}}$ ).
    }
  }
}
}
}
}
}

```

```

/* Phase 4: Evaluation Pass Multipole Operator Build */
for each finest-level cube  $j = 1$  to  $8^D$  {
  if (Length( $p_{D, j}$ ) >  $(l + 1)^2$ ) {
     $L2P(D, j, D, j) = \text{Calculate\_Local\_To\_Potential}(D, j, D, j)$ .
    Add_To_List(evaluation_list,  $p_{D, j}, L2P(D, j, D, j), L_{D, j}$ ).
  }
}
}

```

```

/* Phase 5: Preconditioner Operator Build */
for each finest-level cube  $j = 1$  to  $8^D$  {
    matrix = Lookup_Potential_Coefficients( $D, j, D, N(D, j)$ ).
    Convert_Dielectric_Rows(matrix).
     $\bar{C}(D, j) = \text{Delete\_Rows}(j, N(D, j), \text{Inverse}(\text{matrix}))$ .
    Add_To_List(preconditioner_list,  $q_{C,j}, \bar{C}(D, j), q_{N(D,j),j}$ ).
}

```

Algorithm 4.2 uses several variables that are particular to this implementation. The list below gives the variables in the order they appear, omitting several variables that are defined in the capacitance calculation algorithm (Algorithm 4.1) description.

`direct_list` is the list of destination vector, transformation matrix, source vector triples used to perform all the direct (Coulomb's Law) interaction required by the multipole algorithm.

`matrix` is a temporary matrix variable.

`preconditioner_list` is the list of destination vector, transformation matrix, source vector triples used to apply the preconditioner to the current charge vector iterate.

`up_list` is the list of destination vector, transformation matrix, source vector triples used to convert panel charge to a complete hierarchical set of multipole expansions.

`identity` is the identity matrix. This matrix never explicitly multiplies any vector; it is used as a flag to copy a multipole-expansion pointer.

`level` is a temporary level number variable.

`cube` is a temporary cube number variable.

`down_list` is the list of destination vector, transformation matrix, source vector triples used to convert the multipole-expansion hierarchy into a local-expansion hierarchy.

`evaluation_list` is the list of destination vector, transformation matrix, source vector triples used to evaluate the contribution of the local expansion hierarchy to the evaluation point potentials.

All the functions used by the multipole algorithm initialization function `Initialize_Multipole()` (Algorithm 4.2) are listed here in order of appearance. Appendix B describes the computation done in the transformation-matrix building functions, except for the potential-coefficient calculations for Q2P matrix entries which are described in Section 3.1.

`Build_Cube_Hierarchy(panel_list, depth, o, l)` builds the hierarchy of non-empty cubes, returning the parent and child sets,  $F$  and  $C$  respectively, to allow references to individual

cubes, and the partitioning depth  $D$ . The function also allocates the global total-panel-charge vector,  $q$ , and the vector of evaluation-point potentials,  $p$ , and associates each cube with the contiguous subvectors of  $q$  and  $p$  corresponding to their panels and evaluation points. Each dielectric panel leads to at least 20 off-panel potential evaluation points for the calculation of divided-difference approximations to the electric field at the panel (see also Appendix C and Section 3.3.2). If depth is *auto*, the number of partitioning levels,  $D$ , is set to the smallest number that gives at least 90% of the finest-level cubes' evaluation-point vectors with  $(l + 1)^2$  or less entries. Otherwise  $D$  is set equal to depth.

**Build\_Neighbor\_Sets( $F, C$ )** builds the neighbor cube set,  $N(d, j)$ , for every non-empty cube  $j$  on every level  $2 \leq d \leq D$ .

**Build\_Interaction\_Sets( $F, C$ )** builds the interaction cube set,  $I(d, j)$ , for every non-empty cube  $j$  on every level  $2 \leq d \leq D$ .

**Calculate\_Potential\_Coefficients( $d, j, \bar{d}, \bar{j}$ )** calculates the Q2P transformation matrix that multiplies the charge vector for cube  $\bar{j}$ , level  $\bar{d}$  to produce the evaluation-point potential vector for cube  $j$ , level  $d$ .

**Look\_Up\_Potential\_Coefficients( $d, j, \bar{d}, \text{cube\_set}$ )** looks up previously calculated potential coefficients for all pairwise interactions between panel charges in all the cubes in  $\text{cube\_set}$  on level  $\bar{d}$  and evaluation-point potentials in cube  $j$ , level  $d$ . Multiplying the returned matrix by the panel-charge vector gives the evaluation-point potential vector.

**Convert\_Dielectric\_Rows( $\text{matrix}$ )** transforms all the dielectric panel rows in a matrix of potential coefficients to normal electric field difference coefficients. The transformation is a series of row operations involving rows that represent evaluation point potentials used to construct divided-difference estimates of the electric field difference at the panel. This function implements the dielectric-panel row part of step 2 in Algorithm 3.8.

**Add\_To\_List( $\text{list}, \text{vector\_out}, \text{matrix}, \text{vector\_in}$ )** adds an entry to  $\text{list}$  indicating that one step of the multipole algorithm multiplies  $\text{matrix}$  by  $\text{vector\_in}$  and stores the result in  $\text{vector\_out}$ .

**Set\_Local\_Expansion\_Flag( $d, j$ )** sets the local expansion flag for cube  $j$  on level  $d$ .

**Inverse( $\text{matrix}$ )** converts  $\text{matrix}$  to its inverse in place by Gaussian elimination.

**Delete\_Rows( $j, \text{cube\_set}, \text{matrix}$ )** interprets  $\text{matrix}$  as a matrix of inverse potential and normal electric field coefficients which gives the vector of panel charges contained in  $\text{cube\_set}$ 's cubes when multiplied by the same region's evaluation-point vector, whose entries may have the units of potential or electric field after **Convert\_Dielectric\_Rows()**. The returned matrix is  $\text{matrix}$  with all rows corresponding to charge vector entries for

panels that are outside of lowest-level cube  $j$  removed. This function implements the later part of step 2 in Algorithm 3.8.

**Calculate\_Charge\_To\_Multipole**( $d, j, \bar{d}, \bar{j}$ ) calculates the Q2M transformation matrix that multiplies the charge vector for cube  $\bar{j}$ , level  $\bar{d}$  to produce part of the multipole-expansion coefficient vector for cube  $j$ , level  $d$ .

**Calculate\_Multipole\_To\_Multipole**( $d, j, \bar{d}, \bar{j}$ ) calculates the M2M transformation matrix that multiplies the multipole-expansion coefficient vector for cube  $\bar{j}$ , level  $\bar{d}$ , to produce part of the multipole-expansion coefficient vector for cube  $j$ , level  $d$ .

**Adaptive\_Parent\_Cube**( $d, j$ ) returns the level and cube index of the first ancestor of cube  $j$ , level  $d$ , that has more than one non-empty child cube. The returned cube is called the adaptive parent cube of cube  $j$ , level  $d$ .

**Length**(vector) is the number of entries in vector.

**Size**(set) is the number of elements in set.

**Calculate\_Local\_To\_Local**( $d, j, \bar{d}, \bar{j}$ ) calculates the L2L transformation matrix that multiplies the local expansion-coefficient vector for cube  $\bar{j}$ , level  $\bar{d}$ , to produce part of the local expansion-coefficient vector for cube  $j$ , level  $d$ .

**Calculate\_Multipole\_To\_Local**( $d, j, \bar{d}, \bar{j}$ ) calculates the M2L transformation matrix that multiplies the multipole-expansion coefficient vector for cube  $\bar{j}$ , level  $\bar{d}$ , to produce part of the local expansion-coefficient vector for cube  $j$ , level  $d$ .

**Calculate\_Charge\_To\_Local**( $d, j, \bar{d}, \bar{j}$ ) calculates the Q2L transformation matrix that multiplies the charge vector for cube  $\bar{j}$ , level  $\bar{d}$  to produce part of the local expansion-coefficient vector for cube  $j$ , level  $d$ .

**Local\_Expansion\_Flag\_Set**( $d, j$ ) returns 1 if cube  $j$  on level  $d$  has had its local expansion flag set by **Set\_Local\_Expansion\_Flag**(), that is if the cube has a local expansion.

**Calculate\_Local\_To\_Potential**( $d, j, \bar{d}, \bar{j}$ ) calculates the L2P transformation matrix that multiplies the local expansion-coefficient vector for cube  $\bar{j}$ , level  $\bar{d}$  to produce part of the evaluation point potential vector for cube  $j$ , level  $d$ .

**Calculate\_Multipole\_To\_Potential**( $d, j, \bar{d}, \bar{j}$ ) calculates the M2P transformation matrix that multiplies the multipole-expansion coefficient vector for cube  $\bar{j}$ , level  $\bar{d}$  to produce part of the local expansion-coefficient vector for cube  $j$ , level  $d$ .

### 4.1.3 Multipole Algorithm Field Calculation

After building the geometric-dependent multipole transformation matrices using `InitializeMultipole()`, the vector of panel potentials and flux-density differences,  $p$ , can be calculated given a vector of panel charges,  $q$ , using `CalculateMultipoleFields(p, q)` (Algorithm 4.3). Algorithm 4.3 multiplies the resulting  $p$  by the preconditioner  $\tilde{C}$ , thus performing an approximate matrix-vector product for use in the preconditioned iterative loop of Algorithm 4.1. Algorithm 4.3 is much simpler than `InitializeMultipole(p, q)` (Algorithm 4.2) since it amounts to just performing the matrix-vector products specified by the lists `preconditioner_list`, `direct_list`, `up_list`, `down_list` and `evaluation_list`. Each list entry specifies a precomputed transformation matrix, the vector it multiplies, and where to store the result. In Phase 2 of the calculation, the assignment `vector_out = vector_in` is implemented as a copying of a pointer to `vector_in` to a pointer to `vector_out`, rather than copying the actual vector. The pointer assignment is assumed to have no cost in the complexity estimates of Section 4.3.

The calculation uses two local vector variables, `vector_out` and `vector_in`, the temporary matrix variable `matrix`, and the temporary panel variable `panel`. Sections 4.1.1 and 4.1.2 describe other, previously defined variables.

The multipole algorithm field calculation function uses two new functions. `ApplyPreconditioner(q)` premultiplies  $q$  by the preconditioner matrix using the previously computed `preconditioner_list`. The function is also used before the GMRES loop in the top-level capacitance calculation algorithm, Algorithm 4.1, to multiply the right-hand-side by the preconditioner. The vectors  $q^{out}$  and  $q^{in}$  are subvectors of the panel-charge vector  $q$ .

```
/* Pre-Multiply the Charge Vector by the Stored Preconditioner */
for each  $q^{out}, \tilde{C}, q^{in}$  triple in
  preconditioner_list {
     $q^{out} = \tilde{C} \cdot q^{in}$ .
  }
```

`CalculateElectricFieldDifference(panel i)` estimates the integral of the electric field difference over panel  $i$  using the method described in Section 3.3.2 and Appendix C. By default the function uses a first-order ( $o = 1$  in Algorithm 4.1) method based on a single evaluation point at panel  $i$ 's centroid, but all the higher-order methods of Appendix C ( $o = 2, 3, 5$  or  $7$ ) are implemented. The function `BuildCubeHierarchy()` in Algorithm 4.2 initializes the panel data structures to reflect the integral approximation order  $o$ , making explicit references to  $o$  unnecessary in this function.

Algorithm 4.3 (`CalculateMultipoleFields(p, q)`)

```
/* Phase 1: Direct Multipole Operator Application */
```

```

For each  $p^{out}$ ,  $Q2P$ , and  $q^{in}$  triple in direct_list {
     $p^{out} = Q2P \cdot q^{in}$ .
}

/* Phase 2: Upward Pass Multipole Operator Application */
For each vector_out, matrix, vector_in triple in up_list {
    if (matrix is the identity) {
        vector_out = vector_in.
    }
    else {
        vector_out = vector_out + matrix · vector_in.
    }
}

/* Phase 3: Downward Pass Multipole Operator Application */
For each vector_out, matrix, vector_in triple in down_list {
    vector_out = vector_out + matrix · vector_in.
}

/* Phase 4: Evaluation Pass Multipole Operator Application */
For each  $p^{in}$ ,  $L2P$  and  $L$  triple in evaluation_list {
     $p^{in} = p^{in} + L2P \cdot L$ .
}

/* Phase 5: Dielectric Panel Electric Field Difference Calculation */
for each dielectric panel  $i$  in panel_list {
     $p_i = \text{Calculate\_Electric\_Field\_Difference}(\text{panel } i)$ .
}

/* Phase 6: Preconditioner Operator Application */
Apply\_Preconditioner( $q$ ).

```

## 4.2 Convergence and Error

The capacitance calculation algorithm examined here is based on solving an algebraic approximation to a system of integral equations using a multipole-accelerated iterative method. The error in the resulting approximate solution to the analytic equations has several sources. Section 4.2.1 treats the integral equation's intrinsic sensitivity to discretization error. Sec-



tion 4.2.2 investigates the closely related error arising from the weighted-residuals discretization, which must be shown to tend to zero at some acceptable rate as the fineness of the discretization is increased. Section 4.2.3 discusses the last class of errors, those resulting from the approximate solution of the discretized problem with a multipole-accelerated iterative method.

### 4.2.1 Uniqueness and Well-Posedness

The charge layer approach to the capacitance problem with arbitrary dielectrics (3.32–3.35) leads to the system of integral equations formed by (3.36) and (3.39),

$$f(x) = \iint_S \sigma(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|_2} dS', \quad x \in S_c; \quad (4.1)$$

$$0 = \frac{\epsilon_a + \epsilon_b}{2\epsilon_0} \sigma(x) - (\epsilon_a - \epsilon_b) \iint_S \sigma(x') \frac{\cos(x - x', n_b)}{4\pi\epsilon_0 \|x - x'\|_2^2} dS', \quad x \in S^{ab}, \quad (4.2)$$

where (4.2) must hold for all dielectric interface surfaces  $S^{ab}$ . The system may be viewed as a single first-kind integral equation coupled with several homogeneous second-kind integral equations, one for each dielectric interface. The first-kind equation is a linear operator from  $\sigma$  onto  $f$  constrained by the second-kind boundary operator equations [72].

The system (4.1–4.2) is a partial formulation of the differential equation problem (3.32–3.35). The differential equation, known as Laplace's equation with mixed boundary conditions, can be shown to have a unique, harmonic solution  $\psi$  whenever the conductor and dielectric surfaces are regular and the specified boundary conditions are continuous [66, 63]. Most conductor surfaces encountered in practice are regular, including those with corners and edges.

The first-kind integral equation (4.1) also admits a unique solution  $\sigma$  for every  $f$  when the surfaces are regular but, unlike a harmonic function,  $\sigma$  need not have any derivatives<sup>3</sup>, even if  $f$  is continuous [126]. This is a general property of first-kind integral equations. The solution is often more poorly behaved than the boundary conditions and small perturbations in the boundary conditions can lead to discontinuous changes in the solution [30]. First-kind equations with this behavior are said to be ill-posed since even small numerical errors in an approximate representation of the boundary conditions could lead to large errors in the solution. For the particular first-kind integral operator (4.1), however, the integrand's singularity makes the operator strongly elliptic [93, 139, 4, 72]. A strongly elliptic operator can be shown to behave reasonably when discretized, thus making the first-kind formulation well-posed for practical purposes [93, 139], although some existing results make smoothness assumptions that preclude conductors with edges [4, 139, 63], and others allow edges, but consider generalized functions

<sup>3</sup>More precisely, using terminology defined elsewhere (for example [32, 34]), the first-kind integral operator is an isomorphism (or bijection) of the Sobolev space  $H^s(S)$  onto  $H^{s+1}(S)$ , for any real  $s$ . If the surface is not regular, the operator is still an isomorphism from  $H^{-1/2}(S)$  onto  $H^{1/2}(S)$  [93, 4].  $H^s(S)$  is the space of tempered distributions  $u$  defined on  $S$  such that  $(1 + |\xi|^2)^{s/2} \hat{u}$  is square summable. Here  $\hat{u}(\xi)$ ,  $\xi \in R^2$ , is the Fourier transform of  $u$ .

as possible solutions [93, 126]. In practice, as Section 4.4 demonstrates, the formulation is adequate, even for conductors with edges.

#### 4.2.2 Weighted-Residuals Discretization Convergence and Error

The error in the weighted-residuals discretization of the potential problem depends on the specific method used. Galerkin methods can be shown to converge to the true solution as the number of finite-element expansion functions is increased. Similar analyses for collocation applied to first-kind integral equations exist only for two-dimensional problems [4, 140]. An extension to three dimensions for the capacitance problem may be possible, but is beyond the scope of this thesis. As the results in Section 4.4 indicate, the collocation-Galerkin discretization applied to the first- and second-kind integral-equation formulation (3.36–3.39) performs well in practice.

All the weighted-residuals methods can lead to large errors due to poor modeling of charge-density singularities at conductor edges [45, 78]. General techniques for reducing the error exist but are applicable mainly to Galerkin methods [11]. In the context of capacitance calculation, other methods, such as requiring all edges to be rounded [145], discretizing more finely near edges [109] or solving for the integral of the charge density [45], are used. A full discussion of these modeling issues is beyond the scope of this thesis, but Chapter 5 gives some additional comments.

#### 4.2.3 Linear System Solution Convergence and Error

The linear system resulting from the weighted-residuals discretization of a first-kind integral equation, unlike the corresponding matrix for a second-kind integral, is not necessarily well conditioned [49, 47, 36]. The numerical instability manifests itself in the limit of a fine discretization. The singularity of the free-space Green's function ensures that the diagonal entry is the largest entry in any row derived from the first-kind integral equation. This weak diagonal dominance leads to well conditioned matrices in most cases [15] but not always, especially for finer discretizations [13]. In contrast, the diagonal element in a row corresponding to the second-kind equation remains constant as the discretization is refined, ensuring stability. The capacitance-problem formulation used in this thesis always includes a first-kind part arising from (3.36), and a second-kind part, from (3.39), whenever dielectric interfaces are present. The matrix arising from the discretization is usually dominated by possibly poorly conditioned first-kind rows corresponding to the conductor panels. However, the use of a preconditioner as described in Section 3.2.4 ensures that the effective system matrix is well conditioned, as the results in Section 4.4 demonstrate.

## Capacitance Error Bounds

The combined effects of matrix conditioning, approximate iterative solution, and multipole-algorithm error on the convergence of multipole accelerated GMRES iterative methods like Algorithm 4.1 can be expressed as bounds on the capacitance errors in terms of the sum of the iterative-loop tolerance and the multipole-algorithm error. The bounds apply to capacitances calculated using a preconditioned linear system of the form

$$\tilde{C}Aq = \tilde{C}b, \quad (4.3)$$

where  $q$  is the panel-charge vector,  $b$  is the vector of conductor-panel potentials combined with zeros for the dielectric panel rows,  $A$  is an  $n \times n$  matrix of potential and normal-electric-field coefficients, and  $\tilde{C}$  is a preconditioner appropriate for  $A$ . In particular, the linear systems formed by the  $A$  or  $A^H$  matrices arising from the collocation or collocation-Galerkin strategies of Section 3.3.2 together with the corresponding preconditioners of Section 3.3.4 lead to capacitance-matrix errors that can be bounded in this fashion.

An  $m$ -conductor capacitance problem requires the solution of  $m$  linear systems of the form (4.3),

$$\tilde{C}Aq_i = \tilde{C}b_i, \quad 1 \leq i \leq m, \quad (4.4)$$

where  $q_i$  and  $b_i$  are the panel-charge and right-hand-side vectors corresponding to conductor  $i$  at unit potential, the others at zero. Adding the entries of  $q_i$  corresponding to panels on each conductor gives  $c_i$ , the  $i$ -th column of the capacitance matrix  $C$ ,

$$C \triangleq \begin{bmatrix} | & | & \cdots & | \\ c_1 & c_2 & \cdots & c_m \\ | & | & \cdots & | \end{bmatrix}.$$

Since  $b_i$  has ones in the entries corresponding to the panels on conductor  $i$ , and zeros otherwise,

$$c_i = B^T q_i, \quad (4.5)$$

where the superscript  $T$  denotes transpose, and

$$B \triangleq \begin{bmatrix} | & | & \cdots & | \\ b_1 & b_2 & \cdots & b_m \\ | & | & \cdots & | \end{bmatrix}.$$

Using an iterative method to solve (4.4) produces an approximation to the exact  $q_i$  whose error is bounded by the properties of the iterative method. In the case of the GMRES iterative method used in Algorithm 4.1, the approximate solution after the  $k$ -th iteration,  $\hat{q}_i^k$ , has the Euclidean norm of its residual error bounded by some tolerance. The link between the capacitances and the panel charges (4.5) relates this tolerance to the error in the capacitance matrix.

If  $\hat{c}_i \triangleq B^T \hat{q}_i^k$ , then using (4.4) to substitute gives

$$\begin{aligned} c_i - \hat{c}_i &= B^T (q_i - \hat{q}_i^k); \\ c_i - \hat{c}_i &= B^T [(\tilde{C}A)^{-1} \tilde{C}b_i - \hat{q}_i^k]; \\ \|c_i - \hat{c}_i\|_\infty &\leq \|c_i - \hat{c}_i\|_2 \leq \|B^T\|_2 \|(\tilde{C}A)^{-1}\|_2 \|(\tilde{C}b_i - \tilde{C}A\hat{q}_i^k)\|_2, \end{aligned} \quad (4.6)$$

where  $\|\cdot\|_\infty$  denotes the  $L_\infty$  norm [50]<sup>4</sup>. The last norm on the right in (4.6) is the norm of the residual corresponding to  $\hat{q}_i^k$  considered as a solution to (4.4).

The use of the multipole-algorithm approximation to compute the GMRES iterates leads to bounds on a different residual than that in (4.6). Since the multipole algorithm is a linear operator (see Section 3.2.3), using the multipole-algorithm approximation effectively replaces  $A$  with  $A + E$ , where  $E$  is a bounded perturbation to  $A$ . Thus the multipole accelerated GMRES iterative method solves the linear systems

$$\tilde{C}(A + E)\bar{q}_i = \tilde{C}b_i, \quad 1 \leq i \leq m, \quad (4.7)$$

rather than (4.4), giving bounds on the residual norm

$$\|\tilde{C}b_i - \tilde{C}(A + E)\bar{q}_i^k\|_2 \quad (4.8)$$

rather than  $\|\tilde{C}b_i - \tilde{C}A\hat{q}_i^k\|_2$ . Rearranging (4.6) and using the triangle inequality uncovers (4.8) and a term that depends only on the multipole error,

$$\|c_i - \hat{c}_i\|_\infty \leq \|B^T\|_2 \|(\tilde{C}A)^{-1}\|_2 \left[ \|(\tilde{C}b_i - \tilde{C}(A + E)\bar{q}_i^k)\|_2 + \|\tilde{C}E\hat{q}_i^k\|_2 \right]. \quad (4.9)$$

Taking the norm of (4.4) gives

$$\|\tilde{C}A\|_2 \|q_i\|_2 \geq \|\tilde{C}b_i\|_2,$$

which combined with (4.9) leads to a relative error bound

$$\frac{\|c_i - \hat{c}_i\|_\infty}{\|q_i\|_2} \leq \|B^T\|_2 \kappa(\tilde{C}A) \left[ \frac{\|(\tilde{C}b_i - \tilde{C}(A + E)\bar{q}_i^k)\|_2}{\|\tilde{C}b_i\|_2} + \frac{\|\tilde{C}E\hat{q}_i^k\|_2}{\|\tilde{C}b_i\|_2} \right], \quad (4.10)$$

where  $\kappa(\tilde{C}A) = \|(\tilde{C}A)^{-1}\|_2 \|\tilde{C}A\|_2$  is the condition number of  $\tilde{C}A$  [50]. Since the multipole accelerated GMRES iterative method decreases the normalized residual in (4.10) on every iteration, it can be reduced to any specified tolerance,  $GMRES_{tol} \geq 0$ , so that

$$\frac{\|(\tilde{C}b_i - \tilde{C}(A + E)\bar{q}_i^k)\|_2}{\|\tilde{C}b_i\|_2} \leq GMRES_{tol}$$

Substitution into (4.10) using this expression and (4.4) gives

$$\frac{\|c_i - \hat{c}_i\|_\infty}{\|q_i\|_2} \leq \|B^T\|_2 \kappa(\tilde{C}A) \left[ GMRES_{tol} + \frac{\|\tilde{C}E\hat{q}_i^k\|_2}{\|\tilde{C}Aq_i\|_2} \right]. \quad (4.11)$$

<sup>4</sup>For an  $n$ -entry vector  $v$ ,  $\|v\|_\infty = \max_{1 \leq i \leq n} |v_i|$ . For a  $p \times n$  matrix  $A$ ,  $\|A\|_\infty = \max_{1 \leq i \leq p} \sum_{j=1}^n |A_{ij}|$ .

The normalization of the capacitance matrix error can be made more attractive by assuming that the discretization is fine enough for all the panel charges on a given conductor to have the same sign. The panel charge vector can be written as the sum of two vectors,

$$q_i = q_{iQ} + q_{iD}, \quad (4.12)$$

where  $q_{iQ}$  is  $q_i$  with all the dielectric-panel charge entries set to zero, and  $q_{iD}$  is  $q_i$  with all the conductor-panel charge entries set to zero. The sign pattern assumption constrains  $q_{iQ}$  such that

$$\|q_{iQ}\|_2 \leq \|B^T q_{iQ}\|_2 = \|B^T q_i\|_2 = \|c_i\|_2.$$

Substituting this into the triangle inequality applied to (4.12) gives

$$\|q_i\|_2 \leq \|c_i\|_2 + \|q_{iD}\|_2.$$

This result, together with

$$\|B^T\|_2 \leq \sqrt{\|B^T\|_1 \|B^T\|_\infty} = \sqrt{\max_{1 \leq i \leq m} n_i},$$

where  $\|\cdot\|_1$  denotes the  $L_1$  norm [50]<sup>5</sup> and  $n_i$  is the number of panels on the  $i$ -th conductor, gives the final error bound when substituted into (4.11),

$$\frac{\|c_i - \tilde{c}_i\|_\infty}{\|c_i\|_\infty} \leq \sqrt{\max_{1 \leq i \leq m} n_i} \frac{\|c_i\|_2 + \|q_{iD}\|_2}{\|c_i\|_\infty} \kappa(\tilde{C}A) \left[ GMRES_{tol} + \frac{\|\tilde{C}E\tilde{q}_i^k\|_2}{\|\tilde{C}Aq_i\|_2} \right]. \quad (4.13)$$

### Error Bound Interpretation

The bound is an upper limit for the error in any capacitance associated with any conductor relative to the conductor's self capacitance. The GMRES tolerance bound and the term involving the multipole-algorithm error add since the two corresponding error mechanisms are independent of each other. The error allowed by the GMRES residual bound is equivalent to a right-hand-side perturbation, while the multipole-algorithm error produces a system-matrix perturbation. If the two types of errors are considered independently, bounds similar to the two parts of (4.13) result [127]. For example, GMRES applied to (4.4),

$$\tilde{C}Aq_i = \tilde{C}b_i, \quad 1 \leq i \leq m, \quad (4.14)$$

with no multipole acceleration leads to an error bound analogous to a normalized version of (4.6),

$$\frac{\|q_i - \tilde{q}_i^k\|_2}{\|q_i\|_2} \leq \kappa(\tilde{C}A) \frac{\|(\tilde{C}b_i - \tilde{C}A\tilde{q}_i^k)\|_2}{\|\tilde{C}b_i\|_2}, \quad (4.15)$$

<sup>5</sup>For an  $n$ -entry vector  $v$ ,  $\|v\|_1 = \sum_{i=1}^n |v_i|$ . For a  $p \times n$  matrix  $A$ ,  $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^p |A_{ij}|$ .

where  $\hat{q}_i^k$  is the  $k$ -th iterate approximation to the exact solution  $q_i$ . Also an exact solution of the linear system perturbed by the multipole approximation (4.7),

$$\tilde{C}(A + E)\tilde{q}_i = \tilde{C}b_i, \quad 1 \leq i \leq m, \quad (4.16)$$

has error

$$\frac{\|q_i - \tilde{q}_i\|_2}{\|\tilde{q}_i\|_2} \leq \kappa(\tilde{C}A) \frac{\|\tilde{C}E\tilde{q}_i\|_2}{\|\tilde{C}Aq_i\|_2}, \quad (4.17)$$

as may be verified by subtracting (4.16) from (4.14) taking norms, and dividing by the norm of (4.4). Here  $\tilde{q}_i$  is the exact solution to (4.16).

The bounds (4.15) and (4.17) suggest that the normalized GMRES residual and the multipole-algorithm error should be multiplied by the condition number of the preconditioned matrix in (4.13), as is the case, neglecting the first two terms in the bound. The first two terms occur because (4.13) bounds the capacitance error, rather than the charge error. The leading term,  $\sqrt{\max_i n_i}$ , reflects the fact that even if each entry of the panel charge vector  $\hat{q}_i^k$  differs from the true value in  $q_i$  by no more than  $\delta$ , summing the charges over the conductors can give errors as big as  $\delta \max_i n_i$  in the capacitances. The fraction  $(\|c_i\|_2 + \|q_{iD}\|_2)/\|c_i\|_\infty$  is present because the dielectric-interface-panel charges  $q_{iD}$  must be computed as accurately as the conductor-panel charges, but are not directly used to calculate capacitances. This factor is unimportant for problems with no dielectric interfaces. The condition number  $\kappa(\tilde{C}A)$  is approximately one if the preconditioner is effective.

Thus the quantity in brackets in (4.13) bounds the capacitance matrix error as closely as is possible given the structure of the problem. Furthermore, in most cases the multipole-algorithm error term can be made comparable with  $GMRES_{tol}$  using expansions with fairly low order. In practice, using expansion order  $l$  gives a comparable multipole-algorithm error term when  $GMRES_{tol} = 10^{2-l}$ . The results of Section 4.4 are the most convincing evidence supporting this assertion, but a semi-analytic plausibility argument can be made by examining the multipole-algorithm error term in (4.13),

$$\frac{\|\tilde{C}E\hat{q}_i^k\|_2}{\|\tilde{C}Aq_i\|_2}$$

If  $\tilde{C}$  is normal, a good approximation in practice, then it has an orthogonal set of eigenvectors so  $\|\tilde{C}\|_2 = 1/\|\tilde{C}^{-1}\|_2$  and

$$\frac{\|\tilde{C}E\hat{q}_i^k\|_2}{\|\tilde{C}Aq_i\|_2} \leq \frac{\|\tilde{C}\|_2 \|E\hat{q}_i^k\|_2}{\|\tilde{C}Aq_i\|_2} = \frac{\|E\hat{q}_i^k\|_2}{\|\tilde{C}^{-1}\|_2 \|\tilde{C}Aq_i\|_2} \leq \frac{\|E\hat{q}_i^k\|_2}{\|Aq_i\|_2}$$

Assuming the iterative method is near convergence,  $\hat{q}_i \approx q_i$  so that the multipole-algorithm error term is approximately bounded by

$$\frac{\|E\hat{q}_i^k\|_2}{\|A\hat{q}_i^k\|_2}, \quad (4.18)$$

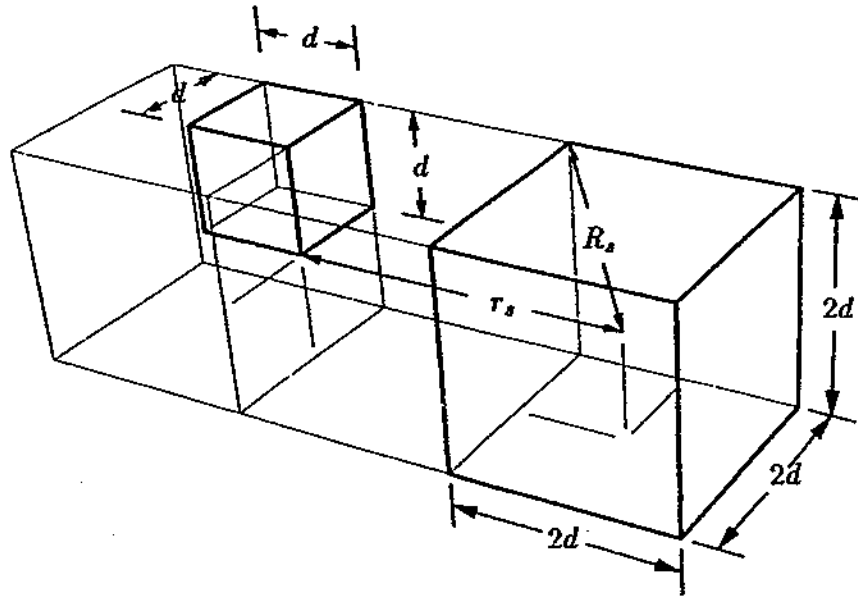


FIGURE 4-1: The cube geometry leading to the minimum separation distance,  $r_s - R_s$ , between charge and evaluation points for a cube hierarchy with  $d \times d \times d$  lowest-level cubes.

the relative error in the multipole-algorithm fields calculated using the current iterate.

A bound on (4.18) in terms of the expansion order  $l$  would allow bounds on the capacitance error in terms of  $GMREStol$  and  $l$  through substitution in (4.13). Such bounds are lacking for the general, multiple-dielectric case where the multipole-algorithm computes both electric fields and potentials. For homogeneous dielectric problems the bound

$$\frac{\|E\hat{q}_i^k\|_\infty}{\|\hat{q}_i^k\|_1} 4\pi\epsilon_0(r_s - R_s) \leq (1/\sqrt{3})^{l+1} \quad (4.19)$$

applies, where  $l$  is the expansion order,  $\epsilon_0$  is the permittivity of free space, and  $r_s$  and  $R_s$  are the values of the corresponding quantities  $r$  and  $R$  in Figures 3-1 and 3-4 that give the smallest separation distance,  $r_s - R_s$ , allowed by the spatial-partitioning hierarchy [52]. The hierarchy also guarantees that  $R/r < 1/\sqrt{3}$  for all expansion evaluations. Combining these facts with the Q2M and Q2L error bounds, (3.7) and (3.11) respectively, and similar bounds for the remaining transformations gives (4.19). Figure 4-1 illustrates that  $r_s - R_s = (3 - \sqrt{3})d$ , where  $d$  is the side length of the hierarchy's lowest-level cubes. The smallest separation distance occurs when a lowest-level cube has a second-nearest neighbor of its parent in its interactions set<sup>6</sup>. Since  $r_s - R_s$  is a lower bound on the separation distance between any charge represented by a truncated multipole or local expansion,

$$\|A\hat{q}_i\|_\infty \leq \frac{\|\hat{q}_i^k\|_1}{4\pi\epsilon_0(r_s - R_s)},$$

<sup>6</sup>The same minimum separation distance and error bound result when the roles of the two cubes are reversed, but the current implementation does not allow interactions of that type.

so that the quantity bounded by (4.19) is actually a lower bound on the relative error of the multipole algorithm measured in the  $L_\infty$  norm,

$$\frac{\|E\hat{q}_i^k\|_\infty}{\|\hat{q}_i^k\|_1} 4\pi\epsilon_0(r_s - R_s) \leq \frac{\|E\hat{q}_i\|_\infty}{\|A\hat{q}_i\|_\infty},$$

making upper bounds on (4.18) difficult to derive using (4.19). The existence of (4.19), however, suggests that the multipole-algorithm error can be made comparable to  $GMRES_{tol}$  using reasonable values of  $l$ . In practice, using expansion order  $l$  gives a comparable multipole-algorithm error term when  $GMRES_{tol} = 10^{2-l}$ . The results in Section 4.4 give further evidence of this.

### 4.3 Algorithmic Complexity

The multipole accelerated capacitance calculation algorithm generally requires nearly  $O(mn)$  time and  $O(n)$  storage for a  $m$ -conductor problem discretized with  $n$  panels, regardless of the details of the discretization method. This Section gives bounds on the time and memory complexity for the particular preconditioned adaptive multipole accelerated GMRES method implementation (PAMA) described in Section 4.1, but it has sufficient generality to apply to closely related methods, for example those using a fully Galerkin discretization or block iterative techniques.

Table 4-1 summarizes the bounds on the time and memory costs for the complete capacitance calculation algorithm, and Table 4-2 gives a detailed account of the multipole algorithm costs for an  $m$ -conductor problem discretized with  $n$  panels. Repeated passes through the main GMRES loop of Algorithm 4.1 calculate new iterates, stopping when the residual is small enough. For an  $m$ -conductor problem with  $n$  panels, the new iterate calculation begins with an  $O(n)$  application of the multipole algorithm to calculate a new Krylov space vector, followed by the construction of the new Krylov-subspace basis including the new Krylov vector. Referring to Algorithm 4.1, the  $k$ -th basis-update step performs  $k$  inner products between the first  $k$   $v^j$  vectors and the new Krylov vector, and then uses the results to orthogonalize the new vector against the known  $v^j$ 's at a cost of  $k\pi$  multiply-adds. The norm computation and normalization steps require  $n$  multiply-adds each, while the residual computation costs  $4(k-1)$  multiply-adds [112]. Thus the total cost of one pass through the GMRES iterative loop is the  $O(n)$  cost of the multipole algorithm plus the overhead  $2(k+1)n + 4(k-1)$ . If the number of iterations is bounded above by a fairly constant number, independent of problem size and geometry, then the basis-update step requires nearly  $O(n)$  computation, making the total cost of one pass through the loop nearly  $O(n)$ . Under the same assumption of nearly constant iteration count, the complete GMRES linear-system solution gives one column of the  $m \times m$  capacitance matrix in nearly  $O(n)$  time. The outer loop performs  $m$  such solutions, making the complete capacitance matrix solve time requirements nearly  $O(mn)$ , since all the other initialization post-processing procedures are also use  $O(n)$  time and memory. The only significant memory



Operation	Time	Storage	Reasoning
Adaptive multipole algorithm initialization.	$M_t K_t$	$K_k(2n-1) + 2n + M_t$	See Table 4-2.
Application of preconditioner to initial residual.	$125K_c^2 mn$	—	The preconditioner retains at most $K_c$ rows from the inverse of an at most $125K_c \times 125K_c$ submatrix of $A$ to represent the neighbor-cube interactions for the panels in each lowest-level cube and its at most 124 neighboring cubes, and each lowest-level cube contains at least one panel.
Adaptive multipole algorithm field calculation.	$(M_t + 2K_p n) K_f m$	—	Each call multiplies all the matrices constructed by the multipole algorithm initialization at a cost of $M_t$ . In addition, the electric-field post-processing step requires $K_p n$ two-term divided differences.
Krylov subspace basis construction.	$2K_f(K_f + 1)mn + 4K_f(K_f - 1)m$	$(K_f + 1)n$	See explanation in text. The memory is reused for different conductor iterations.
Panel-charge summation.	$K_f mn$	—	Each panel charge is summed on every iteration.
Capacitance symmetrization.	$m^2/2$	—	Off-diagonals are averaged.

$K_c$  is the maximum number of panels in a lowest-level cube.

$K_f$  is the maximum number of GMRES iterations required to achieve the desired accuracy. In practice the preconditioner makes  $K_f$  nearly constant over all problems.

$K_k$  is the memory required for one cube data structure.

$K_t$  is the time required to calculate the most costly of all the transformation matrix entries.

$K_p$  is the number of electric field evaluation points associated with each dielectric-interface panel, which increases with the dielectric-interface-panel Galerkin-integral approximation order,  $o$  in Algorithm 4.1.

$M_t$  is  $n[250K_c^2 + (1 + K_p)(l + 1)^2 + 2002(l + 1)^2] - 4(l + 1)^4$ , the largest possible number of double precision numbers required to store all the transformation matrices (see Table 4-2).

Table 4-1: Multipole accelerated capacitance-calculation algorithm (Algorithm 4.1) time (in multiply-adds) and storage cost (in double precision numbers) bounds. Input, output, allocation and assign times are taken as zero.

Operation		Time	Storage	Reasoning
Cube hierarchy construction.		—	$K_k(2n-1) + 2n$	Assuming all the non-adaptive parent cubes' multipole expansions (for example the dashed-circle expansion in Figure 4-2) are stored in the cube from which they originated, the number of non-empty cubes in the hierarchy is at most the number of nodes in the parent-child translation graph. The bounds follow from (4.21) with (4.20) substituted. The extra $2n$ is for the panel charge and field evaluation vectors ( $q$ and $p$ in Algorithm 4.2).
Neighbor construction.	Q2P	$125K_c^2 K_t n$	$125K_c^2 n$	Each lowest-level cube has Q2P interactions with at most 124 neighbors and itself, and each cube contains at least one panel (Phase 1 in Algorithm 4.2).
Lowest-level construction	Q2M	$(l+1)^2 K_t n$	$(l+1)^2 n$	Each panel leads to one Q2M matrix column (first part of Phase 2, Algorithm 4.2).
Upward pass parent-child translation construction.		$2(n-1)(l+1)^4 K_t$	$2(n-1)(l+1)^4$	See Section 4.3.1. All M2M and Q2M transformation matrices involving coarser than lowest-level cubes are at most $(l+1)^2 \times (l+1)^2$ (second part of Phase 2, Algorithm 4.2).
Downward pass parent-child translation construction.		$2(n-1)(l+1)^4 K_t$	$2(n-1)(l+1)^4$	See Section 4.3.1. All the L2L and L2P transformation matrices that involve coarser than lowest-level cubes are at most $(l+1)^2 \times (l+1)^2$ (part of Phase 3, Algorithm 4.2).
Interaction translation construction.		$1998(l+1)^4 K_t n$	$1998(l+1)^4 n$	See Section 4.3.2. All the M2L, Q2L and Q2P matrices arising from operations specified in interaction lists are no larger than $(l+1)^2 \times (l+1)^2$ (part of Phase 3, Algorithm 4.2).
Lowest level construction	L2P	$(l+1)^2 K_p K_t n$	$(l+1)^2 K_p n$	Each panel leads to $K_p$ L2P matrix columns (Phase 4, Algorithm 4.2).
Preconditioner construction		$(125K_c)^3 K_t n$	$125K_c^2 n$	See second entry in Table 4-1 (Phase 5, Algorithm 4.2).

Table 4-2: Adaptive multipole algorithm initialization (Algorithm 4.2) time (in multiply-adds) and storage cost (in double precision numbers) bounds. Allocation and assign times are taken as zero. Constants are defined in Table 4-1.

requirements of the GMRES loop is storage for the  $K_f + 1$   $n$ -entry  $v^j$  vectors, assuming at most  $K_f$  iterations are required. The nearly constant iteration count assumption ensures that the required memory is nearly  $O(n)$ . In practice this assumption is realistic since the preconditioner typically assures rapid convergence (see also Section 4.4).

Table 4-2 lists the  $O(n)$  costs of the multipole algorithm by computation type, rather than following the parts of the implementation of Section 4.1 exactly. Section 4.3.1 bounds the costs of the parent-child translation operations performed between cubes of the upward and downward passes, the Q2M/M2M and L2L/L2P work, respectively, while Section 4.3.2 derives the bounds on interaction operations, the M2L, Q2L and Q2P transformations associated with the interaction-list entries. Unlike earlier treatments [53, 52, 19], the arguments of Sections 4.3.1 and 4.3.2 assume no bounds on the spatial partitioning hierarchy, following [85]. The lowest-level multipole-expansion build (Q2M) and local-expansion evaluation (L2P) operations, as well as the neighbor cube Q2P interactions are not included in either of these categories and therefore have separate entries in Table 4-1. Together these bounds show that the multipole initialization and field-calculation functions, Algorithms 4.2 and 4.3, require  $O(n)$  time and memory.

#### 4.3.1 Parent-Child Translation Operation Complexity

The bounds for the time and memory required to perform all the upward pass M2M or Q2M operations between child cubes and their adaptive parent cubes (given by `Adaptive_Parent_Cube()` in Algorithm 4.2) are the same as for the downward pass L2L and L2P operations between the adaptive parent cube and its children. The correspondence is ensured by the Identity matrix and the `Set_Local_Expansion_Flag()` function of Algorithms 4.2 and 4.1.3. These mechanisms ensure that cubes with single child cubes use one of their descendants' expansions, and adaptive parent local expansions contribute directly to their first descendant that either has more than one child or is a lowest-level cube.

The cube hierarchy, considering only adaptive parent and lowest-level cubes, leads naturally to a graph that gives the bounds for the parent-child translation operations. The graph is constructed from nodes representing the non-empty lowest-level cubes and all the cubes that are adaptive parent cubes, with edges connecting every cube to its adaptive parent cube, except for the node representing the level 0 cube. This parent-child computation graph is a tree with certain properties. The number of non-empty lowest-level cubes in the hierarchy is denoted  $s$ .

**Lemma 4.1 (Tree Structure of Computation Graph)** *The parent-child computation graph is a tree with  $s$  childless or leaf nodes, and every non-leaf node, except the root, has at least three edges connected to it. That is, each non-leaf node except the root has degree strictly greater than two, while the root has degree at least two.*

*Proof.* A connected graph with  $e$  edges and  $v$  nodes with  $e = v - 1$  is a tree. The parent-child

computation graph is connected because any two nodes in the graph correspond to two cubes that are parts of at least one larger cube. A path from any particular node to another node can always be found passing through the node corresponding to the larger cube, or directly between the two given node's cubes if one of the two given cubes is contained in the other.

Furthermore, every node represents one cube with a single edge connecting it to its adaptive parent, except the node corresponding to the level 0 cube. Since the level 0 node has no such edge,

$$e = v - 1. \tag{4.20}$$

Thus the parent-child translation computation graph is a tree, and the level 0 node may be taken as its root.

Since each non-leaf node in the graph corresponds to an adaptive parent cube, which must have at least two child cubes by definition, every non-leaf node has degree at least two. A non-leaf node that is not the root has degree strictly greater than two, due to the connection to the node representing its adaptive parent. The root node has no such connection, making its degree at least two. The leaves of the upward-pass tree correspond to the  $s$  lowest-level cubes since they are the only cubes that have no children.  $\square$

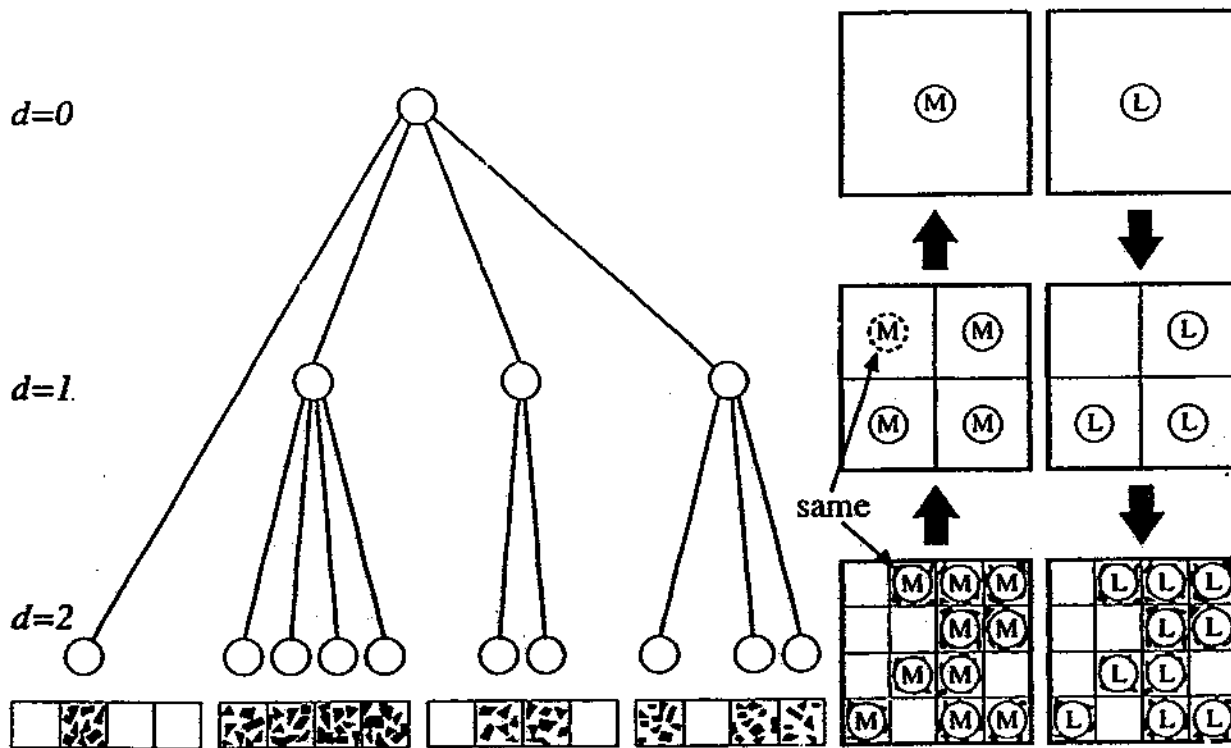


FIGURE 4-2: A parent-child translation computation graph (left) and the corresponding 2-D panel distribution and expansion hierarchies.

Figure 4-2 gives the parent-child translation computation graph for an example charge distribution. As the figure illustrates, each node in the graph corresponds to a cube in the hierarchy

that requires the construction of a multipole expansion and a local expansion. Each edge in the graph represents either an upward pass M2M or Q2M operation converting representations of lower-level charge into higher-level multipole expansions, or an analogous downward pass L2L or L2P operation used to construct the local expansion hierarchy. The one cube that is not an adaptive parent in this example has the dashed-circle multipole expansion, a copy of the child cube's expansion, and no local expansion. The figure also illustrates the correspondence between the non-empty lowest-level cubes and the graph's leaf nodes.

Thus bounding the number of computation-graph edges,  $e$ , gives bounds for the time and memory requirements for the parent-child translation operations. One such bound is a direct result of Lemma 4.1 and elementary graph theory (see, for example [37, 28]). This theorem leads directly to the parent-child translation cost bounds in Table 4-1.

**Theorem 4.1 (Parent-Child Translation Complexity Bound)** *There are at most  $2(n - 1)$  upward- or downward-pass parent-child translation operations required for the adaptive multipole algorithm applied to an  $n$ -panel problem.*

*Proof.* Since the sum of the degrees of any graph's nodes is equal to twice the number of edges in the graph (each edge connects exactly two nodes), Lemma 4.1 implies

$$s + 3(v - s - 1) + 2 \leq 2e,$$

in the case of a parent-child translation computation graph with  $v$  nodes total,  $s$  leaf nodes and  $e$  edges. The first term is the sum of the leaf node degrees, the second corresponds to the non-leaf nodes, and the last is the minimum degree of the root. Since  $e = v - 1$  by Lemma 4.1,

$$e \leq 2(s - 1).$$

All the non-empty lowest-level cubes contain at least one panel so  $s \leq n$  and

$$e \leq 2(n - 1). \tag{4.21}$$

This result and the correspondence between each computation-graph edge and a parent-child translation operation imply Theorem 4.1.  $\square$

### 4.3.2 Interaction-Operation Complexity

Since interaction operations, those M2L, Q2L and Q2P operations involving cubes specified in interaction sets, relate fairly distant panels, they lack the neat parent-child structure of the other translation operations discussed in the previous section. Instead of the tree structure, the interaction-operation count relies on the properties of clustered charge distributions of the type illustrated in two dimensions by Figure 4-3. Such distributions have special properties summarized by Lemma 4.2. Lemma 4.2 assumes that all the interaction set cubes for a given cube are on the same level as the given cube, as do all the results of this section. The implementation of

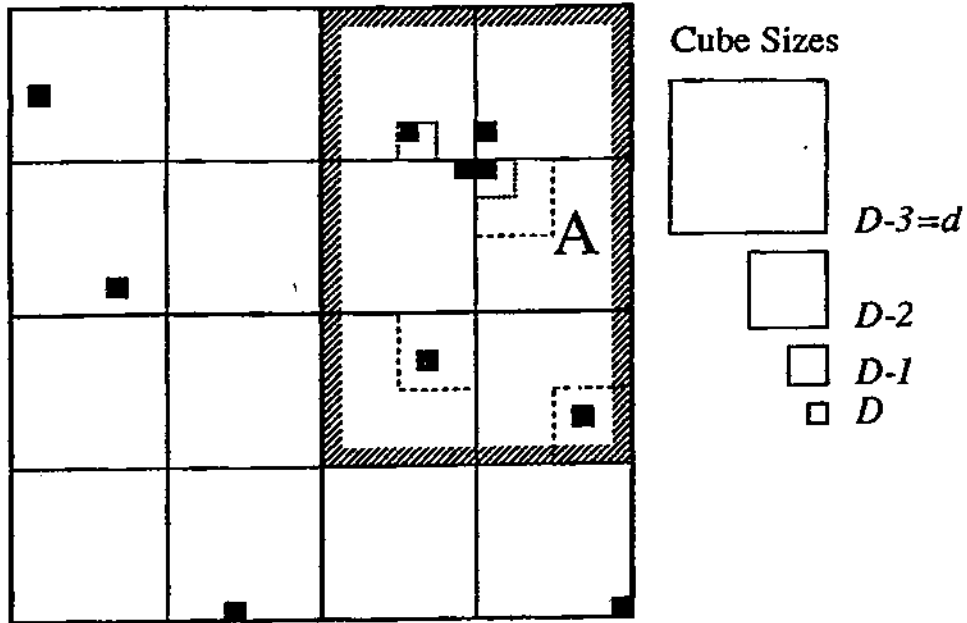


FIGURE 4-3: An example clustered charge distribution. Dark squares indicate filled lowest-level cubes; the shaded region is A's neighbor cube set, and the remaining non-empty cubes are in A's interaction set.

Section 4.1 uses more efficient mixed-size interaction sets, so that the uniform-size assumption leads to upper bounds on the implementation's complexity.

**Lemma 4.2 (Clustered Charge Distribution Properties)** *If every non-empty cube on level  $d$  of a cube hierarchy has exactly one non-empty descendent cube on the lowest level,  $D$ , then the total number of cubes in a given level- $d$  cube's interaction and neighbor sets is greater than or equal to the total number of interaction operations between all cubes on levels  $\bar{d}$ ,  $d \leq \bar{d} \leq D$ , contributing to evaluations on the panels in the given cube.*

*Proof.* The lemma follows directly from the assumed structure of the distribution. A given level- $d$  cube's interaction-set cubes lead to interaction operations between level- $d$  cubes, while each of the given cube's neighbor cubes give rise to exactly one interaction operation between level- $(d+1)$ , or finer, cubes. Each of these neighbor-cube interactions is between the largest cube that is a descendant of the given cube (say on level  $\bar{d}$ ,  $d < \bar{d} \leq D$ ) and contains the given cube's non-empty lowest-level cube, and the analogous level- $\bar{d}$  cube for one of the given cube's neighbors. Since each cube involved in an interaction always contains exactly one non-empty lowest-level cube, each non-empty lowest-level cube that is not a descendant of the given cube gives rise to at most one interaction set operation between cubes on level  $d$ ,  $d+1$ , ..., or  $D$ . The number of neighbor cubes is an upper bound on the interaction operations arising from the neighbor cubes since some of the possible interactions can involve neighboring lowest-level cubes. Thus the total number of interaction- and neighbor-cube set entries for a given level- $d$  cube is an upper bound on the number of interaction operations performing evaluations or

constructing expansions for the given cube and all its non-empty descendants.  $\square$

In Figure 4-3 there are ten non-empty level- $(D - 3)$  cubes in cube A's neighbor and interaction sets, but the level- $D$  cube pair corresponding to the level- $(D - 3)$  cube adjacent and to the left of A involves neighboring level- $D$  cubes, bringing the total number of interaction operations associated with cube A's panels down to nine. Outlined cubes inside cube A's neighboring cubes show the size of the cubes involved in the corresponding interaction operations, illustrating how the interactions can occur on level  $d = D - 3$  or any finer level.

Extending the interaction-operation bounds to general distributions requires an inductive argument based on Lemma 4.2.

**Theorem 4.2 (Interaction Complexity Bound)** *There are at most  $1998n$  interaction operations required for the three-dimensional adaptive multipole algorithm applied to an  $n$ -panel problem.*

*Proof.* The interaction operations associated with the cubes on all levels of the hierarchy can be counted by progressively removing panels from the problem after counting the interaction operations associated with the cubes they empty. The removal is done so as to leave clustered charge distributions that allow repeated application of Lemma 4.2.

To obtain a clustered distribution satisfying the assumptions of Lemma 4.2 with  $d = (D - 1)$ , lowest level-cubes are emptied so that each level- $(D - 1)$  cube has at most one non-empty child cube. In three dimensions, any non-empty cube on the lowest level,  $D$ , has at most 875 interaction operations with the level- $D$  members of its interaction set, and is part of no more than 875 other level- $D$  cubes' interaction sets, making the total number of level- $D$  interactions associated with the emptied cubes no more than  $1750n_D^c$ , where  $n_D^c$  is the number of emptied level- $D$  cubes.

Since a given cube in any three-dimensional spatial-partitioning hierarchy has at most 124 neighboring cubes, and no more than 875 interaction list entries, the given cube has at most 999 cubes in its interaction and neighbor sets, and it is part of at most 999 other cubes' interaction and neighbor sets. This fact and Lemma 4.2 imply that if level- $(D - 1)$  cubes are now emptied in the clustered distribution so that each level- $(D - 2)$  cube has only a single non-empty level- $D$  ancestor, then the removed panels are associated with at most  $1998n_{D-1}^c$  interaction operations, where  $n_{D-1}^c$  is the number of level- $(D - 2)$  cubed emptied. The procedure continues, emptying  $n_d^c$  lowest-level cubes from levels  $d = D, D - 1, \dots, 1$ , and stopping after emptying cubes on level 1, leaving the level-0 cube containing a single non-empty level- $D$  cube. Since every emptied cube contains a single non-empty level- $D$  cube,

$$\sum_{d=D}^1 n_d^c \leq s,$$

where  $s$  is the total number of non-empty lowest-level cubes in the original distribution. Thus the total interaction-operation count is bounded,

$$1750n_D^c + \sum_{d=D-1}^1 1998n_d^c \leq 1998s$$

The final result of Theorem 4.2 follows since each non-empty lowest-level cube contains at least one panel, making  $s \leq n$ .  $\square$

## 4.4 Performance

The practical examples of this section demonstrate the accuracy, and time and memory efficiency of the multipole accelerated capacitance calculation algorithm implementations of Section 4.1: the preconditioned adaptive multipole accelerated GMRES iterative method (PAMA), the adaptive multipole accelerated GMRES method (AMA), and the multipole accelerated GMRES iterative method (MA). The three implementations are referred to collectively as FASTCAP. A result is attributed to FASTCAP in cases where that same result is obtained using all three methods. Section 4.4.1 compares calculated capacitances to capacitance values obtained from other sources, demonstrating the method's accuracy, while Sections 4.4.2 and 4.4.3 give examples showing the algorithm's low time and memory costs. All the computations were performed with a Digital Equipment 3000AXP/500 workstation.

### 4.4.1 Accuracy

The capacitances obtained using the Section 4.1 implementation compare well with analytic and measured values, as well as with values calculated by other methods, even for problems with high permittivity regions. In general the capacitances of a given problem have errors of about 1% of the self capacitances when compared to values calculated using the standard Gaussian elimination approach. The errors are affected by the iterative loop tolerance and expansion order as predicted by the error bounds of Section 4.2.3. The results were obtained using order two multipole expansions, centroid collocation, iterative loop tolerance 0.01 and the automatic spatial partitioning depth, unless otherwise stated ( $l = 2$ ,  $o = 1$ ,  $GMRES_{tol} = 0.01$  and  $depth = auto$ , in Algorithm 4.1). The MA and AMA methods generally have the same accuracy as PAMA, unless the problem is poorly conditioned.

### Comparison to Known Values

Table 4-3 compares the capacitances calculated using FASTCAP to analytic results for the spherical problems of Figure 4-4, and to values calculated elsewhere for the cube and plate problems of Figure 4-5. The spherical conductors have radius 1m, and the plate and cube have 1m sides. The dielectric coating for the coated sphere problem is 1m thick and has permittivity



Method	Problem			
	Sphere 768 Panels	Coated Sphere 864 Panels	Cube 150 Panels	Plate 625 Panels
Gauss. Elim.	110.6	148.1	73.26	40.69
FASTCAP	110.5	148.0	73.18	40.71
Other	111.0‡	148.4‡	73.5, 73.4‡	40.54◊

Table 4-3: Capacitance values (in pF) illustrating FASTCAP's accuracy. †By analytic calculation. ‡From [109, 63]. ◊ From [109].

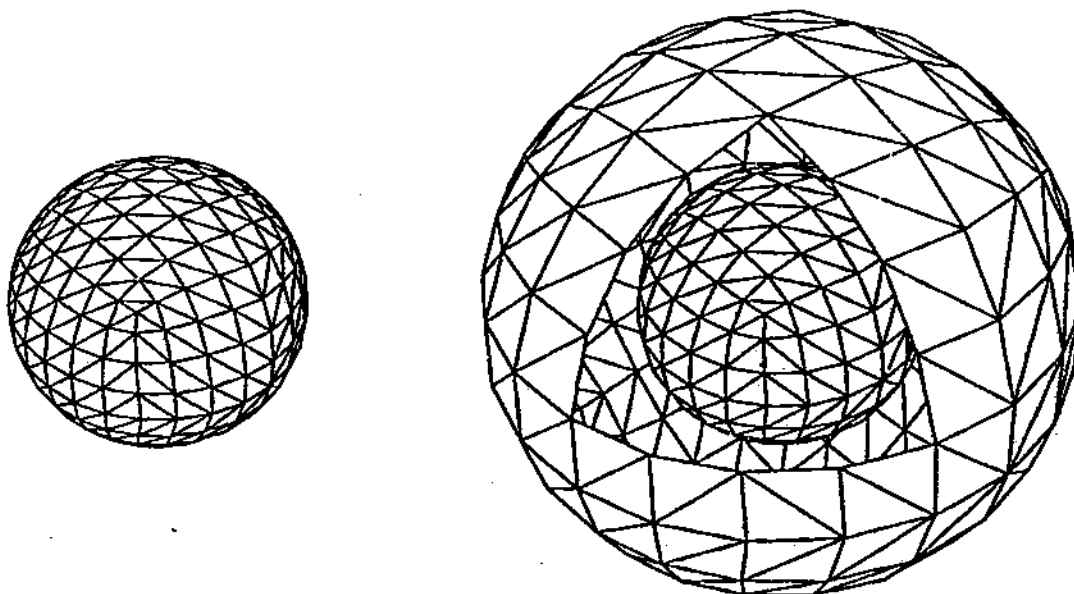


FIGURE 4-4: The sphere (left) and coated sphere (right) discretizations. Some of the dielectric panels have been removed to expose the conductor panels in the coated sphere discretization.

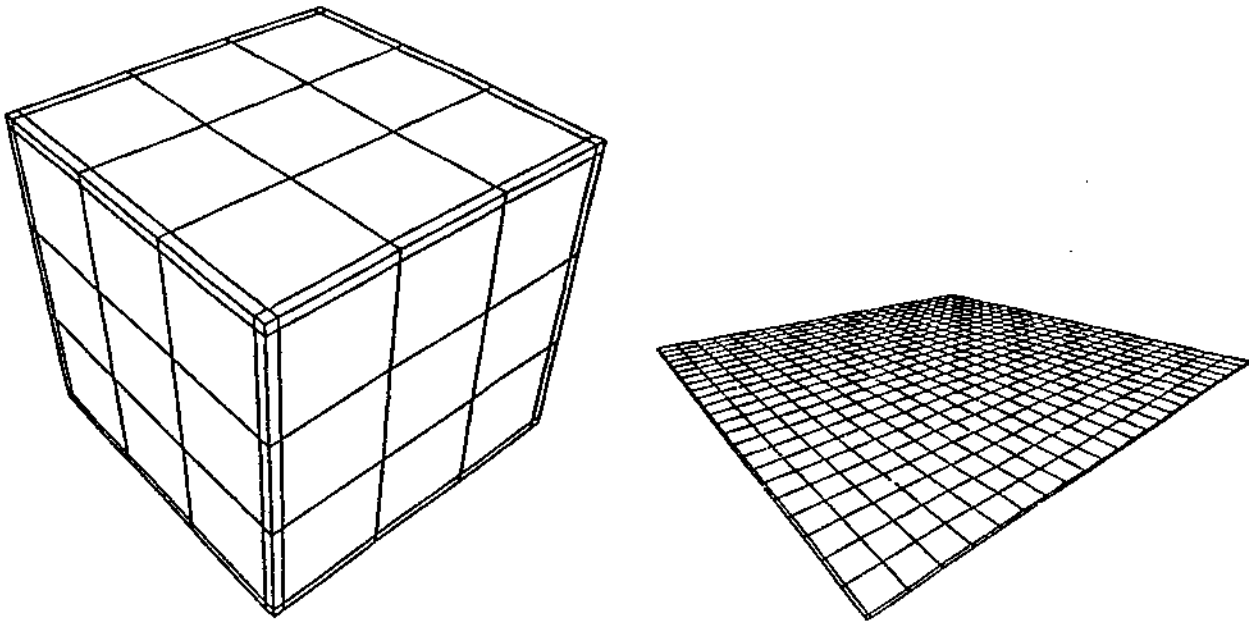


FIGURE 4-5: The cube (left) and plate (right) discretizations.

twice that of free space, the surrounding material for all the examples. The cube faces and plate are discretized using narrow edge panels to better model the charge-density singularity present at conductor edges [109]. Although the plate and cube faces are both  $1\text{m} \times 1\text{m}$ , the stronger edge singularity of the plate's charge density warrants a finer discretization.

The calculated spherical capacitor values are within 1% of the analytic values, while the cube and plate values compare well with other calculations. Also given for comparison are the capacitances obtained using Gaussian elimination applied to the linear system resulting from the same formulation and discretization used by FASTCAP. The close correspondence of these values with the FASTCAP results indicates that the order two multipole approximation is sufficient to solve the linear system with 1% accuracy. Deviations from the true capacitance values by more than 1% are typically caused by discretization error.

### Comparison to Measured Data

The capacitances calculated by FASTCAP compare well with measured values reported for real structures, such as the connector problem in Figure 4-6. Connectors of this type must be analyzed carefully when considered for use in high-speed bus connections [141]. The U-shaped polyester body, with relative permittivity of 3.5, holds sixteen pins with  $0.65\text{mm} \times 0.65\text{mm}$  cross-sections and 3.25mm center to center spacings. Using a 10096-panel discretization, PAMA computes all the self and coupling capacitances for the pins in about twenty CPU minutes. The preconditioner is necessary because the discretization has relatively large panels on the dielectric body, making the corresponding matrix problem poorly conditioned (see also Section 4.4.2). An identical analysis using standard Gaussian elimination algorithms requires roughly 1.5 CPU

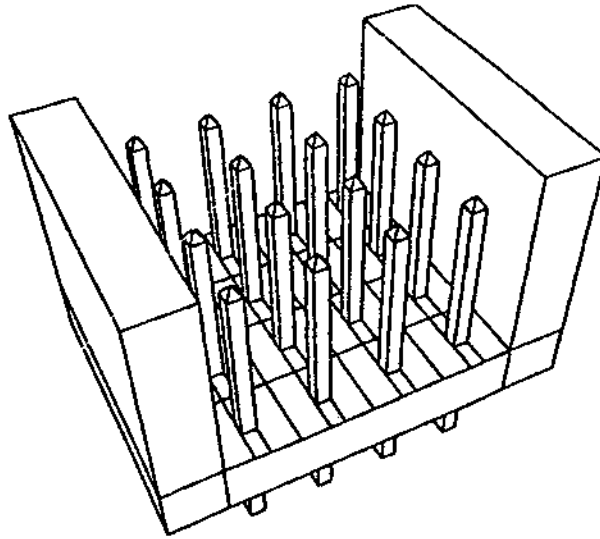


FIGURE 4-6: The backplane connector example. The actual discretization used for the computation is much finer than the one illustrated.

days (see also Table 4-14). The four pins in the center have the highest self capacitances: 0.545pF. The lowest self capacitance is 0.464pF and is attained by the four corner pins. The strongest coupling capacitances, about 0.12pF, occur between corner pins and pins next to the sides of the connector body. By grounding four pins on a main diagonal and the four pins on the remaining parallel two-pin diagonals, the maximum signal pin coupling capacitance is reduced to around 0.071pF. These capacitances compare well with measured values reported in [141].

Figure 4-7 illustrates a second practical example, a simplified model of three adjacent DRAM cells in the 1-Mbit DRAM described in [68]. In Figure 4-7b, the dielectric interfaces are removed to show the conductors more clearly. Each cell consists of a bit line running across the cell, terminating in a via. In Figure 4-7, the three bit lines are elevated on the right side of the figure and run downward to the left where they are attached to the conical vias. The vias connect to the drains of MOS transistors formed by polycide word lines crossing the substrate at right angles to the bit lines and  $0.01 \mu\text{m}$  above the substrate. The polycide word line that controls the transistors is the lower left word line in Figure 4-7. The transistors' sources are all connected to the polycide cell plate, which is also  $0.01 \mu\text{m}$  above the substrate. The dimples in the ground plane below the bit line vias model the capacitors formed by the depletion regions surrounding the drains of the bit cell MOS transistors. There are three other word lines passing through the cells at  $0.7 \mu\text{m}$  (right),  $1.8 \mu\text{m}$  (left) and  $2.2 \mu\text{m}$  (right) above the substrate. All the lines are  $1 \mu\text{m}$  wide and either  $0.3 \mu\text{m}$  (top) or  $0.9 \mu\text{m}$  (bottom) thick. The bit lines are spaced  $2.4 \mu\text{m}$  apart and both sets of word lines are  $2 \mu\text{m}$  apart. The upper, aluminum word lines are covered with a silicon nitride passivation layer with relative permittivity 7.0 and nominal thickness  $0.7 \mu\text{m}$ . The passivation layer is represented by the two dielectric interfaces

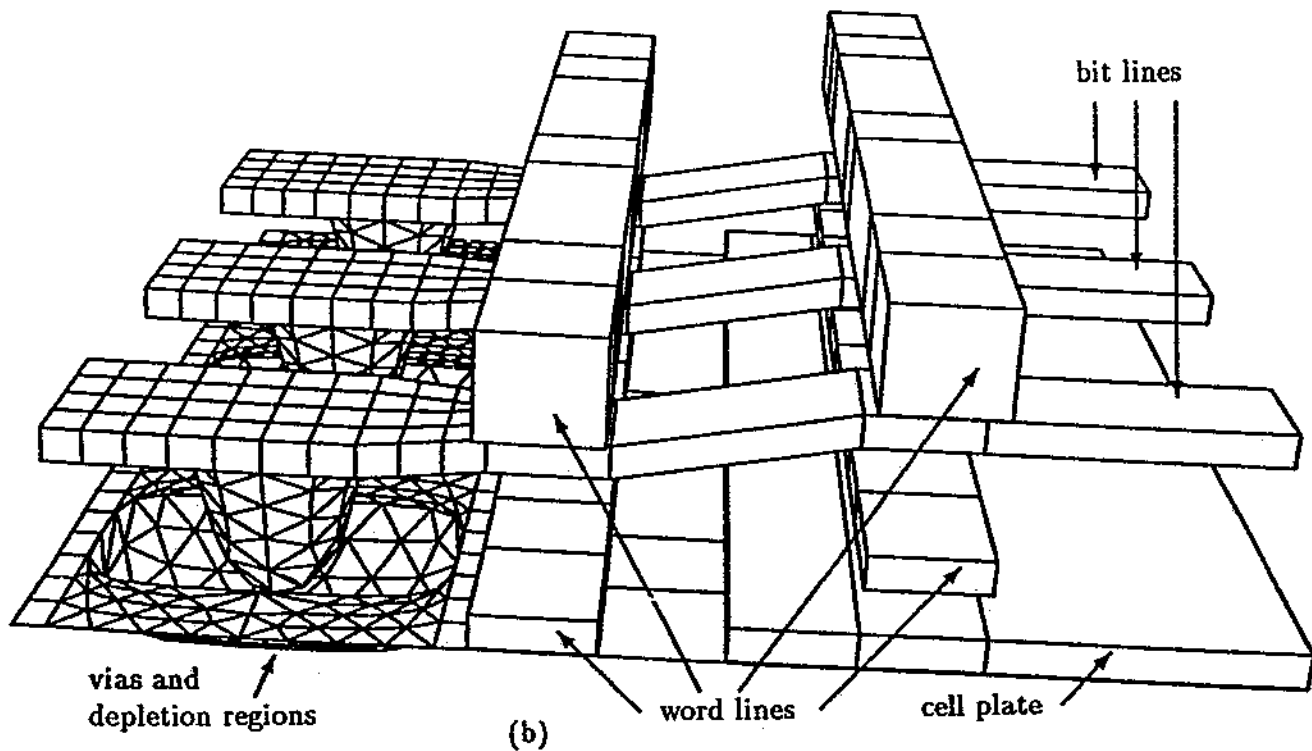
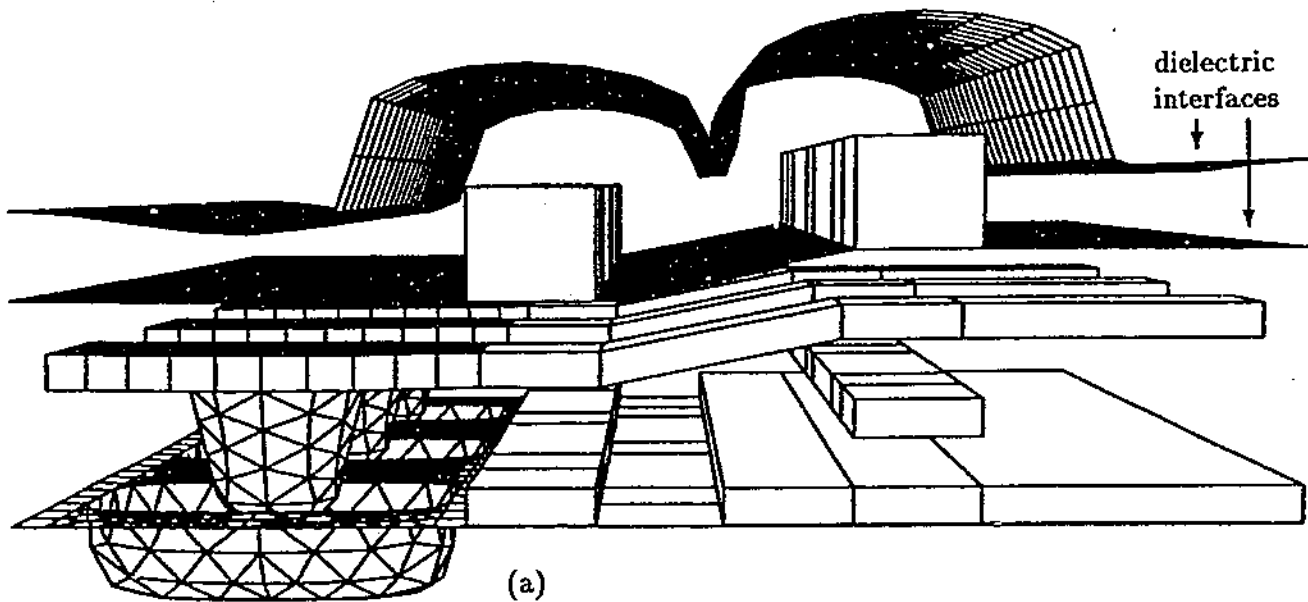


FIGURE 4-7: The complete DRAM model, (a), and with the dielectric interfaces removed for clarity, (b).

illustrated in Figure 4-7a. The material above the top interface is air, while below the lower interface silicon dioxide, with relative permittivity 3.9, is assumed. The computation of the

	near bit line	center bit line	far bit line
near bit line	2.11	-0.063	-0.007
center bit line	-0.063	2.10	-0.063
far bit line	-0.007	-0.063	2.10

Table 4-4: DRAM Bit-line capacitance matrix, fF.

bit-line capacitances given in Table 4-4 using a 6129 panel discretization requires about four minutes using PAMA (see also Table 4-14). The preconditioner is necessary to offset the poor conditioning caused by the closeness of the cell plate and lower polycide word line (see also Section 4-14). The computed capacitances compare well with the measured data given in [68].

The final comparison is between FASTCAP capacitance values and those obtained using formulas based on measured data [21]. The formulas apply to two parallel metal lines over a ground plane, as pictured in Figure 4-8. The line10 case pictured has 10m long lines with

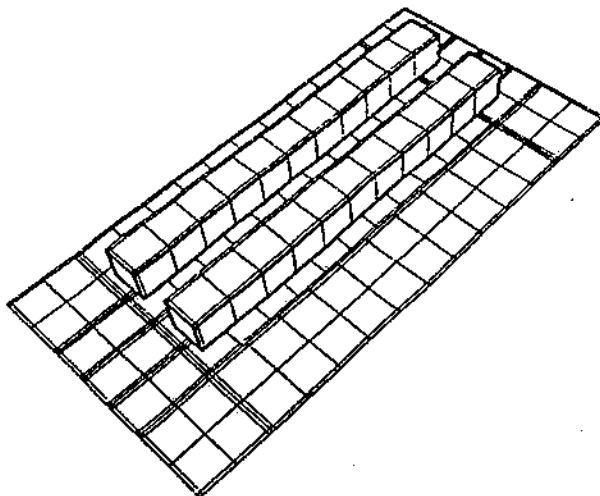


FIGURE 4-8: The line10 discretization of a ten-meter line.

1m×1m cross-sections spaced 1m apart and 1m over the ground plane, which extends 2m beyond the footprints of the lines. Also considered, but not pictured, is line100, which has 100m conductors and 20m ground-plane skirts, but is otherwise identical to line10.

Table 4-5 gives the calculated capacitances for the two cases together with the values derived using the formulas from [21]. The formulas assume the lines have infinite length, so there is more of a discrepancy for the line10 values, where three-dimensional effects are more important. The larger difference in the ground capacitances is probably due to the formulas' difficulty in modeling the fringing-field interaction between the lines and the relatively close ground plane, although the roughly 10% difference is within the rated error of the formulas. The line-to-line

Method	Problem			
	Line10 697 Panels		Line100 4887 Panels	
	line-to-ground	line-to-line	line-to-ground	line-to-line
Formula	167.3	151.0	1673	1510
FASTCAP	264.8	147.5	1884	1587

Table 4-5: Capacitance values (in pF) comparing FASTCAP to the empirically matched formulas of [21].

capacitance, however, is better approximated by the formulas when the lines are close to each other, since their interaction is well approximated by considering only the parallel plates formed by their facing sides, leading to good agreement with the FASTCAP value.

### Comparison to Gaussian Elimination

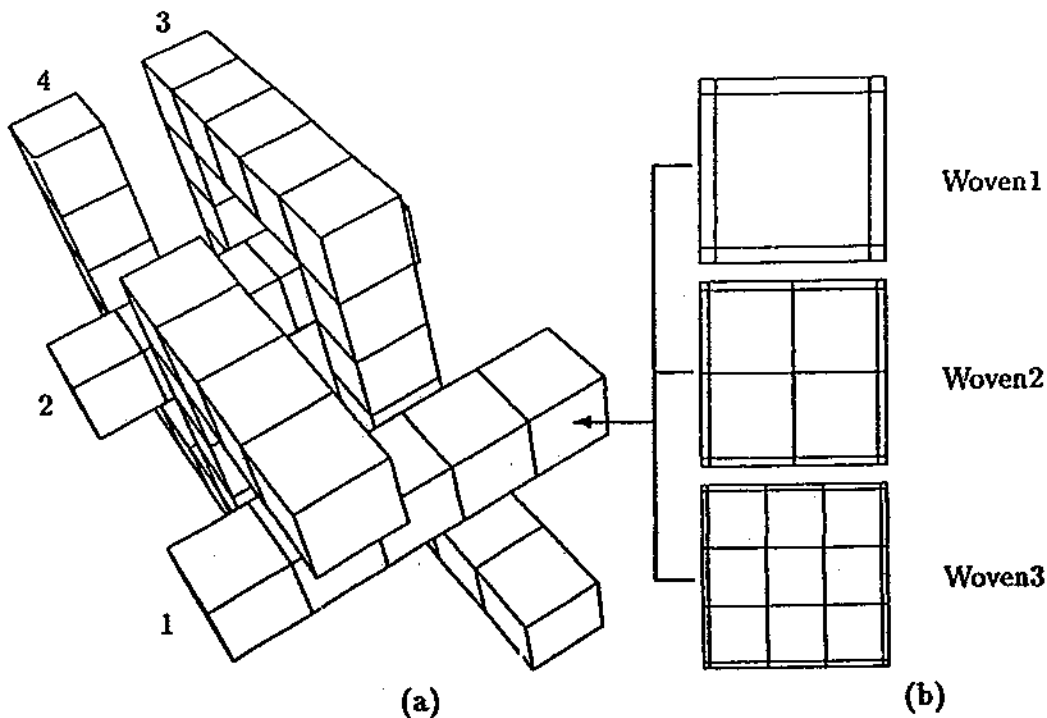
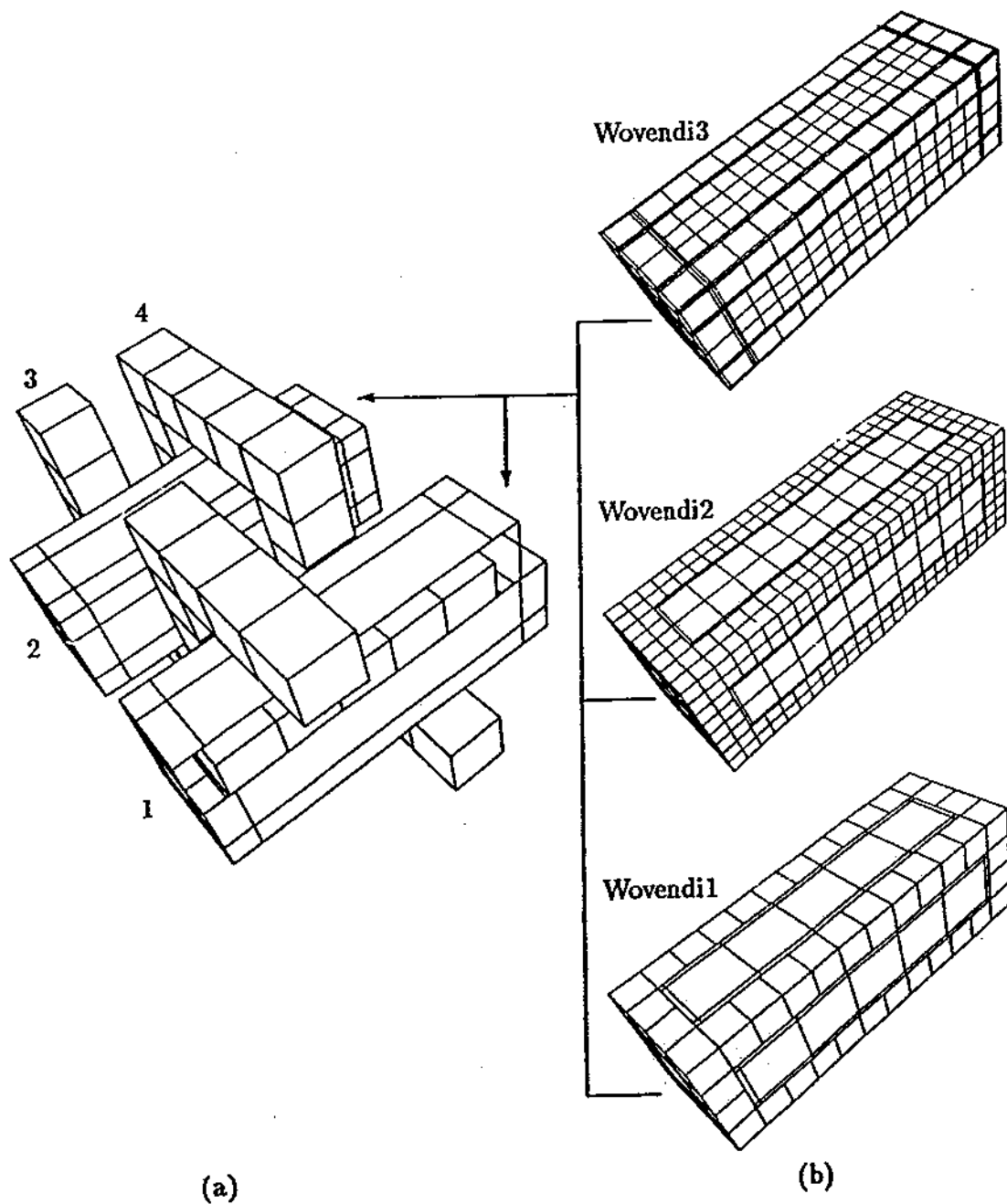


FIGURE 4-9: The  $2 \times 2$  woven-bus problem: bars have  $1m \times 1m$  cross sections. The three discretizations are obtained by replacing each square face in (a) with the corresponding set of panels in (b).

FASTCAP is nearly as accurate as Gaussian elimination even for problems more complicated than those of Table 4-3, such as the  $2 \times 2$  woven-bus structure in Figure 4-9. The capacitances computed using the two methods are compared in Table 4-6, using coarse, medium, and fine



**FIGURE 4-10:** The three  $2 \times 2$  coated woven-bus discretizations are obtained by replacing the two dielectric interfaces in (a) with the corresponding set of panels in (b). Conductor faces are discretized as in Figure 4-9. Part of one dielectric interface has been removed to expose conductor 1 for the illustration.

Method	Problem					
	Woven1		Woven2		Woven3	
	1584 Panels		2816 Panels		4400 Panels	
	$C_{11}$	$C_{12}$	$C_{11}$	$C_{12}$	$C_{11}$	$C_{12}$
Gauss. Elim.	251.6	-6.353	253.2	-6.446	253.7	-6.467
FASTCAP	252.0	-6.572	253.5	-6.793	253.2	-6.316

Table 4-6: Capacitance values (in pF) illustrating FASTCAP's accuracy for the complicated homogeneous-dielectric geometry of Figure 4-9.

Method	Problem					
	Wovendi1		Wovendi2		Wovendi3	
	2060 Panels		4352 Panels		6620 Panels	
	$C_{11}$	$C_{12}$	$C_{11}$	$C_{12}$	$C_{11}$	$C_{12}$
Gauss. Elim.	343.2	-12.13	343.2	-12.13	344.4	-12.24
FASTCAP	344.6	-11.97	344.2	-11.79	347.5	-12.26

Table 4-7: Capacitance values (in pF) illustrating FASTCAP's accuracy for the complicated multiple-dielectric geometry of Figure 4-10.

discretizations of the woven-bus structure, also shown in Figure 4-9. Table 4-7 gives the analogous capacitances for the  $2 \times 2$  coated woven-bus problem in Figure 4-10. The coated problem is identical to the Figure 4-9 geometry, except that the straight conductors are covered with a 0.5m-thick permittivity- $2\epsilon_0$  dielectric. Refining the conductor faces exactly as in Figure 4-9, and using the corresponding dielectric interface discretization illustrated in Figure 4-10 gives coarse, medium and fine discretizations of the coated woven-bus problem.

The coupling capacitance  $C_{12}$  between conductors one and two, which is at least thirty-times smaller than the self capacitance  $C_{11}$  in both woven-bus problems, is computed nearly as accurately with FASTCAP as with Gaussian elimination, and is never in error by more than about 0.1% of  $C_{11}$ . Since  $C_{11}$  and  $C_{12}$  are the smallest self and coupling capacitances, respectively, for both geometries, these results indicate that the multipole algorithm approximation does not interfere with the convergence of the capacitances as the discretization is refined.

### Effect of GMRES Tolerance and Expansion Order

The results of Tables 4-6 and 4-7 use expansion order  $l = 2$  and iterative loop tolerance  $GMRES_{tol} = 0.01$ . As indicated by the error bounds of Section 4.2.3, the error in the capacitances, when compared to the values computed by Gaussian elimination, is bounded above by the sum of two terms, one controlled by  $l$  and the other by  $GMRES_{tol}$ . Table 4-8 and 4-9 show how the capacitance errors for woven1 and wovendi1 follow this trend when FASTCAP's



default expansion order and tolerance change<sup>7</sup>. The errors are measured as percentages of their column's diagonal capacitances to be consistent with the error bounds. In the table headings,  $C_{ij}$  is the value computed by Gaussian elimination and  $\Delta C_{ij}$  is  $C_{ij}$  subtracted from the corresponding value computed by FASTCAP.

<i>GMREStol</i>	Expansion Order, <i>l</i>							
	<i>l</i> = 0		<i>l</i> = 1		<i>l</i> = 2		<i>l</i> = 3	
	$\frac{\Delta C_{11}}{C_{11}}$	$\frac{\Delta C_{12}}{C_{11}}$	$\frac{\Delta C_{11}}{C_{11}}$	$\frac{\Delta C_{12}}{C_{11}}$	$\frac{\Delta C_{11}}{C_{11}}$	$\frac{\Delta C_{12}}{C_{11}}$	$\frac{\Delta C_{11}}{C_{11}}$	$\frac{\Delta C_{12}}{C_{11}}$
$10^{-1}$	0.17%	-1.34%	-1.02%	-0.23%	-0.62%	-0.12%	-0.22%	-0.18%
$10^{-2}$	1.29%	-1.33%	-0.46%	-0.27%	0.17%	-0.09%	0.21%	-0.16%
$10^{-3}$	1.29%	-1.24%	-0.56%	-0.13%	0.11%	0.06%	0.12%	-0.02%
$10^{-4}$	1.29%	-1.23%	-0.55%	-0.12%	0.11%	0.07%	0.13%	-0.01%

Table 4-8: Errors in the largest and smallest capacitances in the woven1 discretization of Figure 4-9 illustrating the effect of expansion order and iterative loop tolerance on FASTCAP's accuracy for problems with a homogeneous dielectric.

<i>GMREStol</i>	Expansion Order, <i>l</i>							
	<i>l</i> = 0		<i>l</i> = 1		<i>l</i> = 2		<i>l</i> = 3	
	$\frac{\Delta C_{11}}{C_{11}}$	$\frac{\Delta C_{12}}{C_{11}}$	$\frac{\Delta C_{11}}{C_{11}}$	$\frac{\Delta C_{12}}{C_{11}}$	$\frac{\Delta C_{11}}{C_{11}}$	$\frac{\Delta C_{12}}{C_{11}}$	$\frac{\Delta C_{11}}{C_{11}}$	$\frac{\Delta C_{12}}{C_{11}}$
$10^{-1}$	11.6%	-2.29%	-1.22%	-0.31%	-1.22%	-0.17%	-1.51%	-0.25%
$10^{-2}$	14.2%	-1.80%	0.41%	-0.13%	0.41%	0.05%	0.18%	-0.07%
$10^{-3}$	14.2%	-1.78%	0.40%	-0.11%	0.39%	0.07%	0.15%	-0.04%
$10^{-4}$	14.2%	-1.79%	0.40%	-0.10%	0.39%	0.09%	0.15%	-0.03%

Table 4-9: Errors in the largest and smallest capacitances in the wovendil discretization of Figure 4-10 illustrating the effect of expansion order and iterative loop tolerance on FASTCAP's accuracy for problems with multiple dielectrics.

In both tables the error produced using decreasing *GMREStol* values decreases to a limiting value for a fixed expansion order. Decreasing the iterative loop tolerance below the point where the limiting error is reached leads to more computation with little decrease in the error. In particular using  $GMREStol = 10^{-l}$  with  $l > 1$  and  $GMREStol = 10^{-2}$  for  $l = 0, 1$  gives the most accurate answers at the lowest cost. Using this strategy leads to  $10^{(2-l)\%}$  error for  $l > 1$ , and between 1% and 10% error for  $l = 0, 1$ . The lack of improvement in  $C_{11}$  when  $l$  changes from 2 to 3 for the woven1 example demonstrates how the discretization's level of refinement ultimately limits accuracy. In the multiple-dielectric problem wovendil, however, the discretization's accuracy limit of about 0.1% is only reached for  $l = 3$ . In general multiple-dielectric problems often require higher order expansions to accurately enforce the derivative

<sup>7</sup>Experiments with  $l < 2$  use *depth=4*, rather than *auto* to avoid excessive spatial partitioning levels.

boundary conditions at the dielectric interfaces.

The limiting behavior of the error as a function of iterative loop tolerance and expansion order is consistent with the form of the error bound (4.13). The results indicate that decreasing the iterative loop tolerance,  $GMREStol$ , while holding expansion order,  $l$ , constant results in decreased error only until  $GMREStol$  reaches a certain value  $\alpha$ , and that  $\alpha$  decreases as  $l$  increases. The form of the bound,  $GMREStol$  plus a term multiplying  $(1/\sqrt{3})^l$ , matches this behavior. The reduction in accuracy for multiple-dielectric problems is also suggested by the factors that weaken (4.13) when dielectric panels are present.

### High Permittivity Regions

$\epsilon_{ratio}$	Dielectric Panel Integral Approximation Order, $o$				
	$o = 1$	$o = 2$	$o = 3$	$o = 5$	$o = 7$
2	-0.24%	-0.58%	-0.64%	-0.71%	-0.04%
4	4.76%	1.45%	0.94%	0.10%	-0.69%
6	0.88%	3.49%	2.44%	1.13%	-0.39%
8	17.4%	5.96%	4.29%	2.17%	-0.05%
10	24.0%	8.45%	6.23%	3.26%	0.25%

Table 4-10: Errors in the computed capacitance for the coated capacitor of Figure 4-12 with  $b = 2$  illustrating the effect of  $\epsilon_{ratio}$  and the dielectric-panel integral approximation order on FASTCAP's accuracy.

The default boundary-element discretization technique used by FASTCAP can lead to appreciable error for problems with high permittivity regions. Figure 4-11 and Table 4-10 characterize the error in the capacitance calculated using the 864-panel discretization pictured on the right in Figure 4-4 for the coated sphere problem illustrated schematically in Figure 4-12. Here the error is the analytic value subtracted from the calculated value, divided by the analytic value. The first column of Table 4-10 demonstrates that the error is a strong function of  $\epsilon_{ratio} \triangleq \epsilon_1/\epsilon_2$  when using the default first-order dielectric-panel Galerkin integral approximation (centroid collocation), while Figure 4-11 shows qualitatively how the error varies with coating thickness when  $\epsilon_{ratio} = 10$ , again using the first-order approximation. Figure 4-11 illustrates that in general the electric field beyond the dielectric interface is always well approximated, but the errors in the potential and the electric field inside the coating and at the interface increase with the coating thickness. The mismatch in the electric fields at the conductor surface, however, remains relatively constant, leading to fairly constant capacitance error as a function of coating thickness. Decreasing the iterative loop tolerance and increasing the expansion order also leave the error essentially unchanged, indicating that the discretization technique, rather than FASTCAP's approximations, causes the error.

Decreasing the error in the electric-field calculations used to enforce Gauss's law on the dielectric-interface panels leads to less error in the electric field inside the coating, and therefore

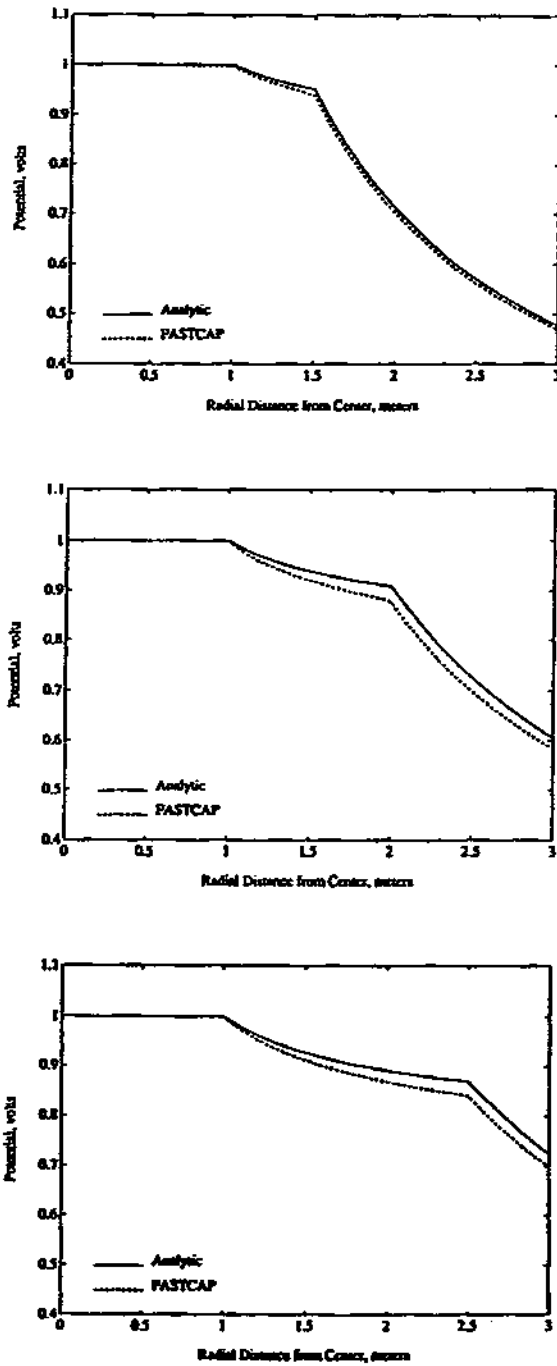


FIGURE 4-11: The analytic and calculated potential as a function of radial distance for the coated capacitor of Figure 4-12 with  $\epsilon_{ratio} = 10$  for  $b = 1.5$  (top),  $b = 2.0$  (middle) and  $b = 2.5$  (bottom).

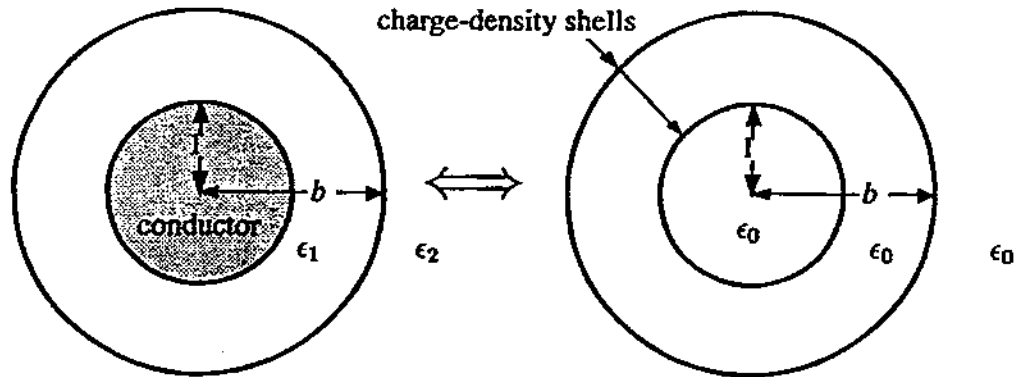


FIGURE 4-12: A schematic drawing of the coated sphere problem and the corresponding equivalent charge representation used by FASTCAP.

more accurate capacitances. Increasing the order of the Galerkin integral approximation used for dielectric panels, refining the discretization or using more complicated expansion functions for the charge density all result in more accurate electric fields. One of these strategies, or some combination of them, is necessary for the equivalent charge formulation used in FASTCAP (see Figure 4-12) because of the disparity in charge-density shells required to model the electric field in this case. In the high-permittivity limit, the electric field in the coating approaches zero, while the outer field becomes asymptotically close to the field of a metal sphere of radius  $b$ . In the absence of discretization error, the smaller field between the shells in the equivalent-charge model is set entirely by the weaker inner charge-density shell. When  $\epsilon_{ratio}$  is large, however, the strong outer-shell charge density's discretization error leads to aberrations in the field between the shells that are of the same order as the weaker inner-shell field. The dominance of the outer-shell discretization error is more complete as the dielectric interface moves farther from the conductor, leading to larger errors in the potential near the interface. In contrast, when  $\epsilon_{ratio} < 1$  the magnitude of the charge carried by both shells approaches the same constant, thereby avoiding the errors.

Total Number of Discretization Panels				
864	1536	2400	6096	10560
24.0%	19.1%	15.7%	11.2%	9.05%

Table 4-11: Errors in the capacitance computed using  $\sigma = 1$  and progressively finer discretizations for the coated capacitor of Figure 4-12 with  $b = 2$  and  $\epsilon_{ratio} = 10$ .

Figure 4-13, and Tables 4-10 and 4-11, show how the capacitance error can be reduced by refining the discretization or increasing the order of the approximate Galerkin dielectric-interface panel integrals,  $\sigma$ , for the geometry of Figure 4-12 with  $b = 2$ . Using the higher-order

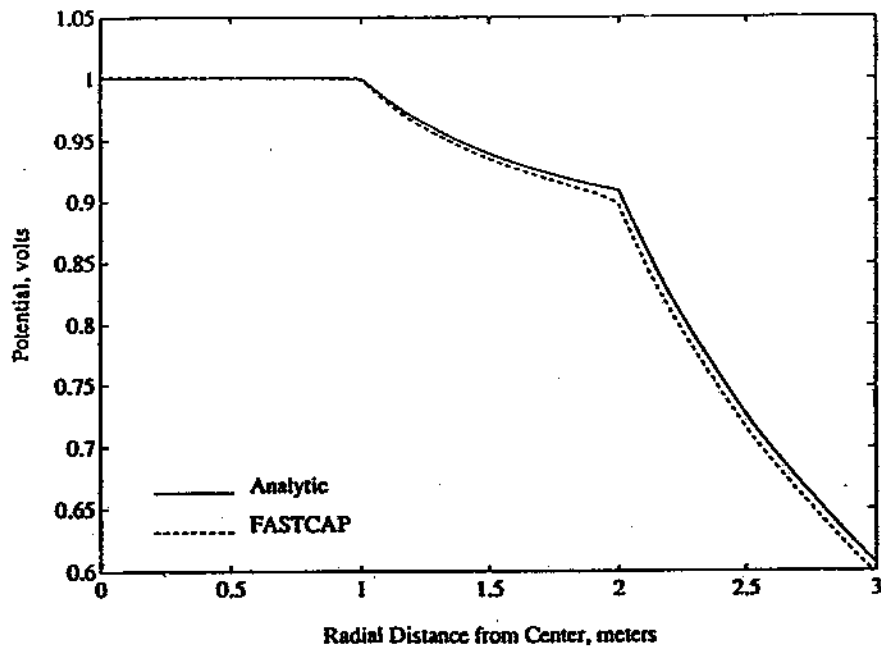
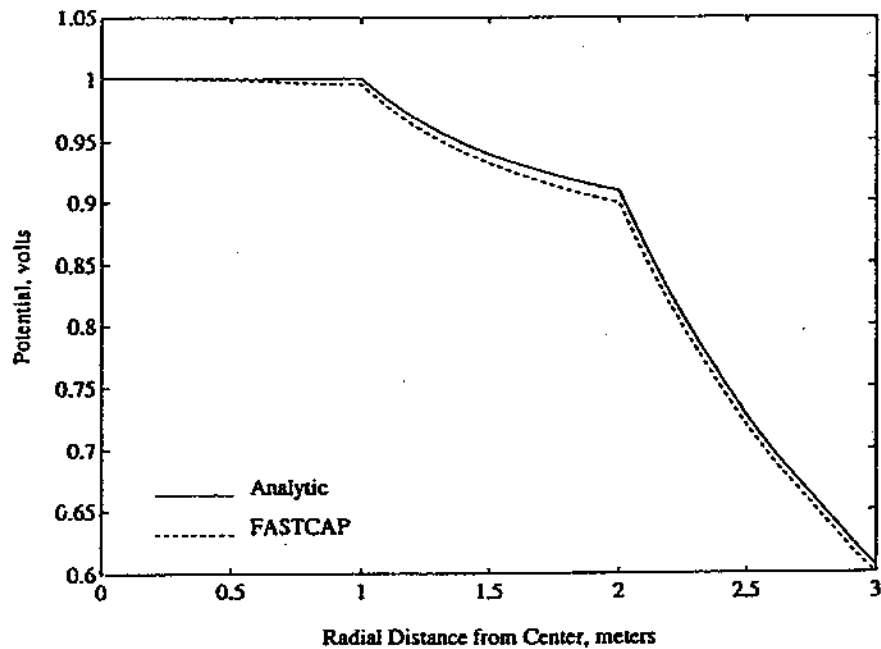


FIGURE 4-13: The analytic and calculated potential as a function of radial distance for the coated capacitor of Figure 4-12 with  $\epsilon_{ratio} = 10$ ,  $b = 2$  using 10560 panels with  $o = 1$  (top), and 864 panels with  $o = 7$  (bottom).

dielectric-interface panel-integral approximations of Appendix C leads to the hybrid collocation-Galerkin approach described in Section 3.3.2 and implemented in FASTCAP. Figure 4-13 shows that both methods reduce the error in the fields, although the highly refined 10560-panel discretization appears to lead to some inaccuracy inside the conductor. This may be the onset of fine-discretization instability due to the first-kind formulation (see Section 4.2). Using the alternative collocation-Galerkin approach gives 1% error up to  $\epsilon_{ratio} = 10$ , as Table 4-10 illustrates.

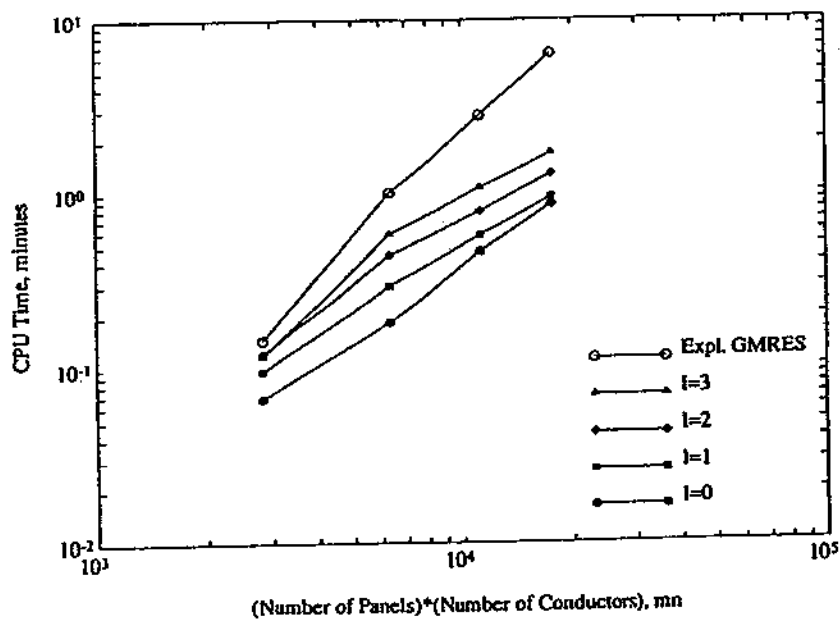
In general, the higher-order collocation-Galerkin methods are only necessary when the charge densities used to model dielectric interfaces in the equivalent-charge problem are much larger in magnitude than the conductor-charge densities. Such a situation can only occur if a conductor is coated with permittivity  $\epsilon_1$  dielectric, with  $\epsilon_1$  much larger than that of the surrounding dielectric,  $\epsilon_2$ . In practice dielectric coatings rarely have  $\epsilon_1/\epsilon_2 > 4$ , so lower-order integral approximations are usually sufficient.

#### 4.4.2 Execution Time

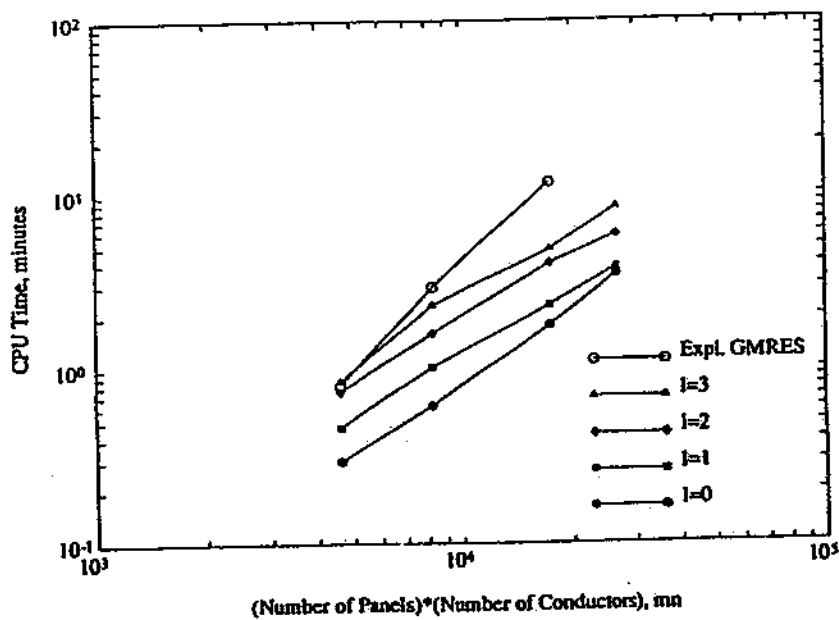
Figure 4-14 gives plots of the total CPU time required by the adaptive multipole accelerated GMRES method (AMA) to calculate the complete  $4 \times 4$  capacitance matrices for the  $2 \times 2$  woven bus and  $2 \times 2$  coated woven bus as a function of the product of the number of panels  $n$ , and the number of conductors  $m$ , in the discretizations of Figures 4-9 and 4-10. A coarser discretization than those illustrated in the figures, with four panels per face, gives the smallest-problem results for each problem. The coarser discretizations are called *woven0* and *wovendi0* for the coated and uncoated woven-bus problems respectively. The four curves in each plot corresponding to the expansion orders  $l = 0, 1, 2$  and  $3$  show that for problems with and without dielectric interfaces, the time complexity is nearly  $O(mn)$ , with the expansion order affecting only the constant that multiplies  $mn$  in the expression for the time<sup>8</sup>. Comparison to the run times of GMRES with explicit, rather than multipole accelerated, matrix-vector products for the same problems indicates that AMA is more efficient for problems having more than roughly one thousand panels.

This performance is typical for well conditioned problems treated with AMA, and for poorly conditioned problems solved using the preconditioned adaptive multipole accelerated algorithm (PAMA). The only exception to this are the  $l = 0$  times, which grow faster than linearly in both of the Figure 4-14 graphs. As the  $l = 0$  iteration counts in Table 4-12 show when compared to the corresponding explicit GMRES counts, the inaccuracy of the order-0 expansions makes the effective matrix problem more poorly conditioned. The higher-order expansions, however, all lead to roughly the same number of iterations as explicit GMRES, indicating that convergence is essentially unaffected by the multipole algorithm approximation when the expansion order

<sup>8</sup>Experiments with  $l < 2$  use *depth = 4* rather than *auto* to avoid excessive spatial partitioning levels.



(a)



(b)

FIGURE 4-14: Execution time in DEC 3000AXP/500 CPU minutes as a function of  $mn$  for AMA and explicit GMRES applied to the woven-bus problem, Figure 4-9, (a) and the coated woven-bus problem, Figure 4-10 (b).

Method	Problem			
	Woven0	Woven1	Woven2	Woven3
	704 Panels	1584 Panels	2816 Panels	4400 Panels
Expl. GMRES	26	50	43	42
AMA, $l = 3$	26	50	43	42
AMA, $l = 2$	26	50	43	43
AMA, $l = 1$	27	52	44	44
AMA, $l = 0$	32	52	49	52

(a)

Method	Problem			
	Wovendi0	Wovendi1	Wovendi2	Wovendi3
	1152 Panels	2060 Panels	4352 Panels	6620 Panels
Expl. GMRES	50	88	77	(76)
AMA, $l = 3$	51	88	78	76
AMA, $l = 2$	51	88	78	76
AMA, $l = 1$	52	88	78	76
AMA, $l = 0$	55	91	82	89

(b)

Table 4-12: The total number of iterations required to solve for the  $4 \times 4$  capacitance matrices of the woven bus of Figure 4-9 (a) and coated woven bus of Figure 4-10 (b) to give the data plotted in Figure 4-14. The parenthesis indicate an extrapolated value for a calculation that needed excessive memory.



$l$  is greater than zero. Table 4-12 also illustrates the variation in iteration counts for different discretizations of the same geometry when the iterative method uses no preconditioner (AMA). The woven1 and wovendi1 discretizations, for example, lead to more iterations than the other discretizations of the same geometry for all the methods.

Method	Problem			
	Woven0 704 Panels	Woven1 1584 Panels	Woven2 2816 Panels	Woven3 4400 Panels
PAMA, $l = 3$	14†	22	20	20
PAMA, $l = 2$	20	22	20	28†
PAMA, $l = 1$	20	23	21	20
PAMA, $l = 0$	24	24	24	24

(a)

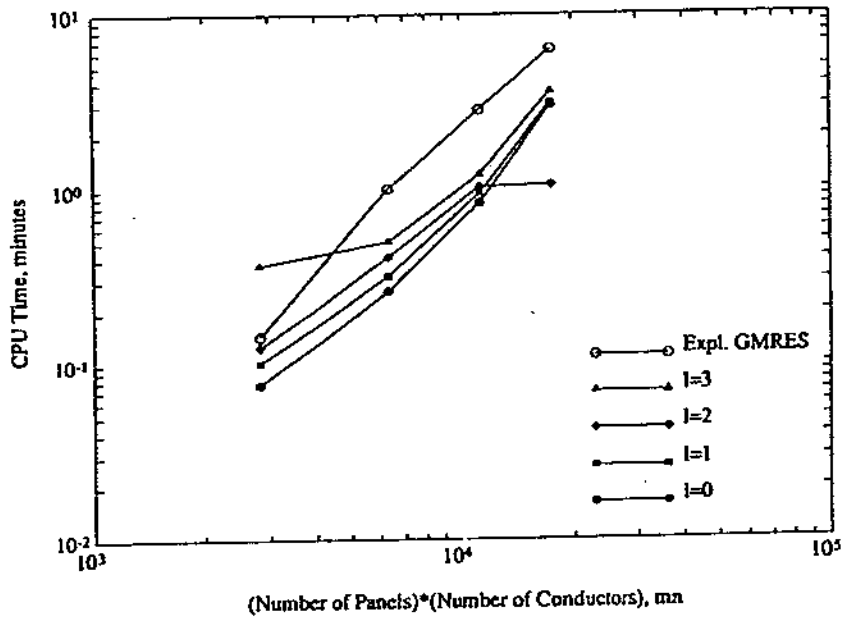
Method	Problem			
	Wovendi0 1152 Panels	Wovendi1 2060 Panels	Wovendi2 4352 Panels	Wovendi3 6620 Panels
PAMA, $l = 3$	23	24	24	32†
PAMA, $l = 2$	23	24	32†	32†
PAMA, $l = 1$	22	24	24	24
PAMA, $l = 0$	24	25	25	25

(b)

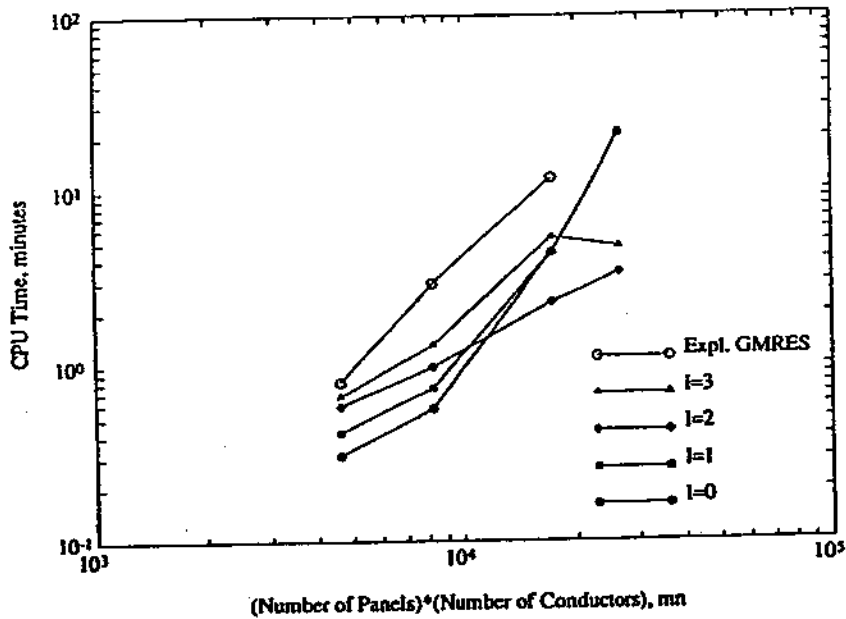
Table 4-13: The total number of iterations required to solve for the  $4 \times 4$  capacitance matrices of the woven bus of Figure 4-9 (a) and coated woven bus of Figure 4-10 (b) to give the data plotted in Figure 4-15. The symbols † and ‡ indicate calculations that use  $D = 5$  and  $D = 3$  spatial partitioning levels respectively; all others use  $D = 4$ .

In contrast, applying PAMA to the same problems results in the reduced, and nearly constant, iteration counts of Table 4-13. Since these example problems are fairly well conditioned, however, the time cost of building and applying the preconditioner is sometimes greater than the savings resulting from reductions in iteration count, as the time plots of Figure 4-15 illustrate when compared to Figure 4-14. In other cases, such as for  $l = 2$  and  $l = 3$  in Figure 4-15, the PAMA time is less than the corresponding AMA time in Figure 4-14. Figure 4-15 makes it clear, however, that inappropriate use of the preconditioner can lead to superlinear growth in run time as a function of problem size.

The variation in the effectiveness of preconditioning is due to differences in the part of the system matrix  $A$  used to compute the preconditioner. A preconditioner that depends on more entries of  $A$  generally is a better approximation to  $A^{-1}$ , leading to a more favorable eigenvalue distribution (see Section 3.2.4), but also higher costs in the construction phase and on each iteration. Since the PAMA preconditioner incorporates all the interactions between the lowest-level cubes and their neighbors (see Section 3.2.4), the fraction of  $A$  used to construct



(a)



(b)

FIGURE 4-15: Execution time in DEC 3000AXP/500 CPU minutes as a function of  $mn$  for PAMA applied to the woven-bus problem, Figure 4-9, (a) and the coated woven-bus problem, Figure 4-10 (b).

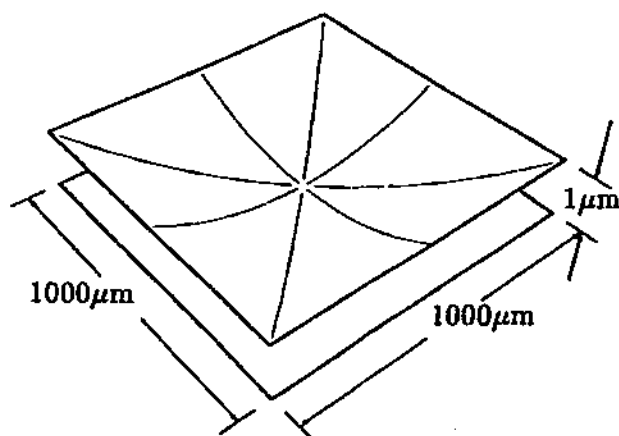


FIGURE 4-16: A schematic illustration of the diaphragm problem. The two plates are  $0.02\mu\text{m}$  apart at the center.

the preconditioner increases as the size of the lowest-level cubes increases. Thus a spatial-partitioning hierarchy constructed for a given discretization yields preconditioners that tend to lower iteration counts as the number of spatial-partition levels,  $D$ , decreases.

As indicated in Table 4-13a, the preconditioner resulting when the spatial-partitioning depth  $D = 3$  is used for the woven-bus problem gives fewer iterations (14) than the  $D = 4$  preconditioner (20–24 iterations), which in turn requires fewer iterations than the  $D = 5$  preconditioner (28). Figure 4-15a shows that the reduced number of iterations required by the  $D = 3$  and  $D = 4$  preconditioners, however, does not offset their increased time costs. For this well conditioned problem, the  $D = 5$  preconditioner is probably best, since it leads to lower execution time in spite of the added iterations it causes. The  $D = 5$  preconditioner is also better for the coated woven-bus problem as the data in Table 4-13b and Figure 4-15b indicate.

For more poorly conditioned problems, the PAMA algorithm usually picks an effective preconditioner that leads to lower execution times. The algorithm's performance could be improved for well conditioned problems such as the woven-bus problems above by using only part, rather than all, of the multipole algorithm's nearby interaction information to form the preconditioner (see also Section 3.2.4). Poorly conditioned problems are more common as the results of Table 4-14 indicate.

Table 4-14 compares execution times for a representative sample of the problems discussed in this section. The new diaphragm problem is a model for the integrated circuit pressure transducer pictured in Figure 4-16 [64]. The  $5 \times 5$  woven bus is the extension of the  $2 \times 2$  woven bus illustrated in Figure 4-17. The table shows that the  $O(mn)$  FASTCAP algorithms (MA, AMA and PAMA) outperform the  $O(n^3)$  Gaussian elimination and  $O(mn^2)$  explicit GMRES methods for problems with a few thousand panels, with the disparity in the algorithms' complexities becoming more apparent as the problem size increases. For moderate sized problems

Homogeneous Dielectric Problems				
Method	Woven2 2816 Panels <i>mn</i> = 11264	Woven3 4400 Panels <i>mn</i> = 17600	Diaphragm 6448 Panels <i>mn</i> = 12896	5 × 5 Woven 9360 Panels <i>mn</i> = 93600
Gauss. Elim.	48.1	175	(585)	(1840)
Expl. GMRES	2.81	6.30	(73.0)	(121)
MA	1.14	2.33	12.2	18.7
AMA	0.80	1.34	4.96	10.5
PAMA	1.06	1.07	0.72	8.11

Multiple Dielectric Problems				
Method	Wovendi1 2060 Panels <i>mn</i> = 8240	Wovendi2 4352 Panels <i>mn</i> = 17408	DRAM 6129 Panels <i>mn</i> = 18387	Connector 10096 Panels <i>mn</i> = 161536
Gauss. Elim.	17.4	185	(508)	(2360)
Expl. GMRES	2.94	11.8	(52.3)	(381)
MA	3.33	11.3	22.8	(155)
AMA	1.62	4.01	8.82	39.0
PAMA	1.02	2.31	3.57	12.1

(a)

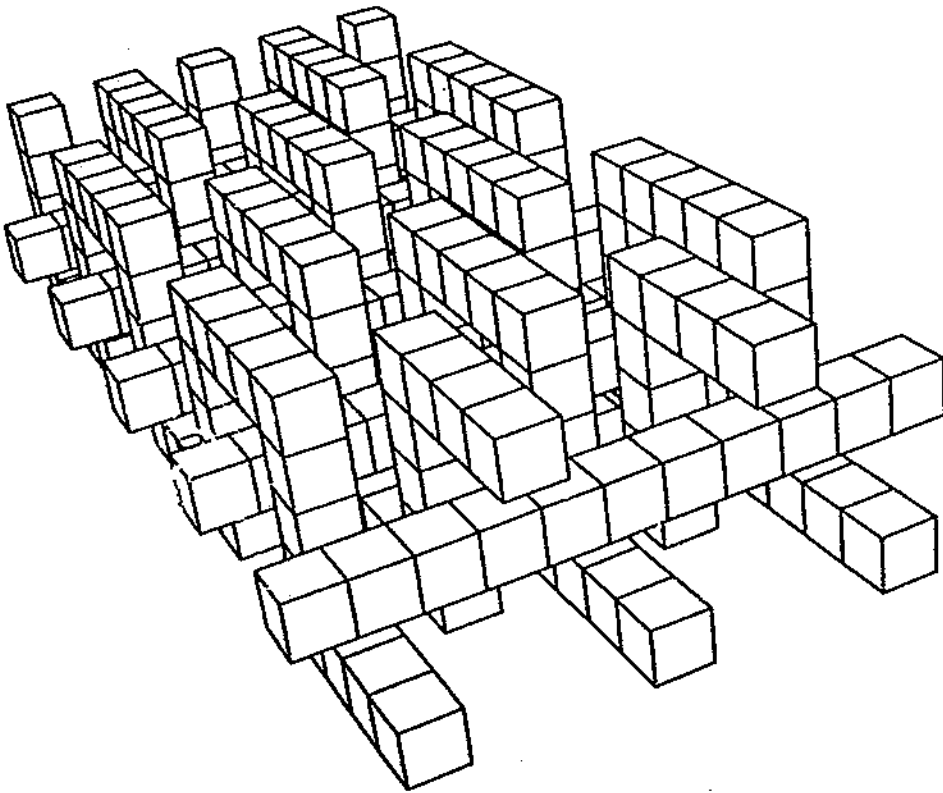
Homogeneous Dielectric Problems				
Method	Woven2 2816 Panels <i>mn</i> = 11264	Woven3 4400 Panels <i>mn</i> = 17600	Diaphragm 6448 Panels <i>mn</i> = 12896	5 × 5 Woven 9360 Panels <i>mn</i> = 93600
Expl. GMRES	43	42	(232)	(183)
MA	43	43	232	183
AMA	43	43	232	183
PAMA	20	28	78	75

Multiple Dielectric Problems				
Method	Wovendi1 2060 Panels <i>mn</i> = 8240	Wovendi2 4352 Panels <i>mn</i> = 17408	DRAM 6129 Panels <i>mn</i> = 18387	Connector 10096 Panels <i>mn</i> = 161536
Expl. GMRES	88	77	(171)	(462)
MA	88	78	171	(462)
AMA	88	78	172	462
PAMA	24	32	51	127

(b)

Table 4-14: The DEC 3000AXP/500 CPU minutes (a) and total number of iterations (b) required to solve for the capacitance matrices of a representative set of problems. Values in parenthesis are extrapolated.



**FIGURE 4-17:** The  $5 \times 5$  woven-bus discretization is obtained by replacing each face in the illustration with nine panels, as for the woven1 problem of Figure 4-9.

in the ten-thousand panel range, PAMA is about two orders of magnitude faster than Gaussian elimination, as the  $5 \times 5$  woven bus and connector results demonstrate.

In general the reduction in computational complexity afforded by the MA method leads to faster run times than for GMRES with explicit multiplications. The MA method's non-adaptive nature, however, causes it to use excessive memory for some problems. The connector problem, for example, could not be solved due to excessive memory use. The poor time performance of the MA method for dielectric problems when compared to the explicit GMRES times is probably due to the current implementation's use of Algorithm 3.6. Algorithm 3.6 requires off-panel evaluation points to calculate electric-field approximations by divided differences. An implementation based on Algorithm 3.5, rather than Algorithm 3.6 avoids off-panel evaluation points, and probably leads to fewer expansions. The AMA method avoids these shortcomings with an adaptive approach and uses about half the CPU time required by the MA method. The final optimization is the preconditioner, which has the most dramatic effect on the poorly conditioned DRAM, diaphragm and connector problems.

A capacitance problem becomes poorly conditioned when the diagonal entries in the rows of the coefficient matrix  $A$  in  $Aq = b$  become small compared to the off-diagonal entries. By Gershgorin's theorem [50], this weakened diagonal dominance can lead to less tightly clustered eigenvalues, implying poorer conditioning and slower GMRES convergence (see also Section 3.2.4). Since the diagonal entry in a given row is a measure of how the charge on a given panel effects the potential or electric field on the given panel, and the off-diagonal entries gauge the influence of other panel charges on the given panel, the diagonal entry can become comparable to the off-diagonal entries if other panels are either very close to, or much larger than the given panel. The DRAM problem, pictured in Figure 4-7, is poorly conditioned primarily because of the close proximity of the cell plate and lower polycide word line to the substrate. The connector problem discretization of Figure 4-6 uses much smaller panels on the conductors than on the large faces of the U-shaped body, leading to poor conditioning that is counteracted by the preconditioner. The most dramatic savings in run time occur when PAMA is applied to the diaphragm problem of Figure 4-16. The discretization uses much larger panels near the edges of the structure than in the center, where the panels from the top and bottom conductors nearly touch. This makes the diaphragm problem the most poorly conditioned problem in Table 4-14, leading to nearly a factor of seven improvement in the PAMA run time compared to the AMA time. In fact, the diaphragm problem is so poorly conditioned that the explicit GMRES, MA and AMA methods produce capacitance matrices that are not diagonally dominant, a physical impossibility. The PAMA result is diagonally dominant, verifying that the preconditioner helps maintain accuracy for poorly conditioned problems, as discussed in Section 3.2.4.

For well conditioned problems like woven2 in Table 4-14 the default preconditioner used by PAMA can lead to increased execution time. In those cases the preconditioner is typically too aggressive and should use a smaller part of the nearby panel interaction information, as

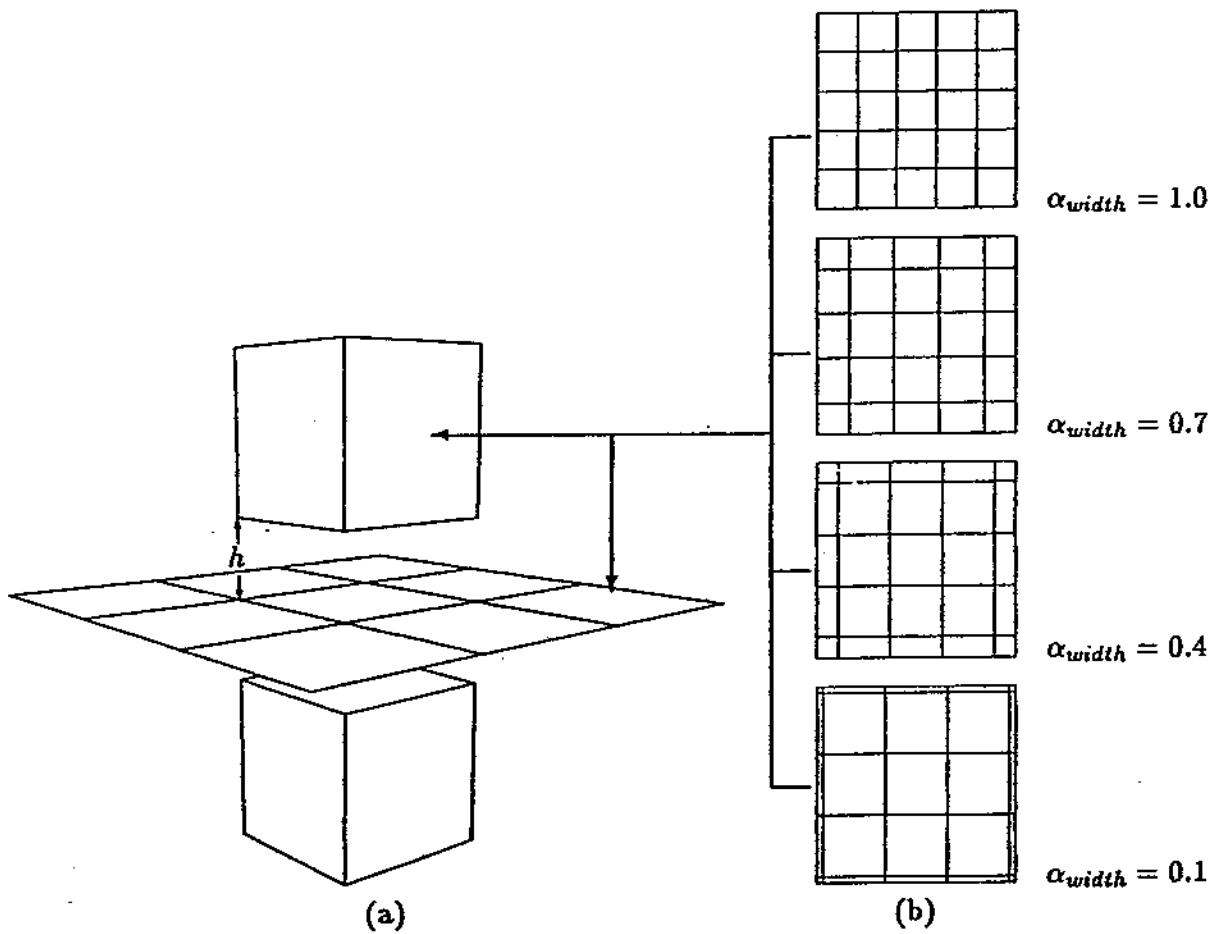


FIGURE 4-18: The sixteen preconditioner test problems leading to the data in Table 4-15 result from varying the distance between the cubes and the dielectric interface,  $h$ , and using the different face discretizations in (b) for all the conductor and dielectric faces shown in (a) to vary the panel edge with ratio,  $\alpha_{width}$ .

AMA CPU Seconds				
$\alpha_{width}$	Distance from Dielectric Interface, $h$ , meters			
	$h = 5 \times 10^{-1}m$	$h = 5 \times 10^{-2}m$	$h = 5 \times 10^{-3}m$	$h = 5 \times 10^{-4}m$
1.0	5.48	6.64	8.14	8.74
0.7	5.46	6.43	9.44	13.4
0.4	5.54	6.86	10.1	13.3
0.1	6.54	7.49	10.9	14.2

(a)

PAMA CPU Seconds				
$\alpha_{width}$	Distance from Dielectric Interface, $h$ , meters			
	$h = 5 \times 10^{-1}m$	$h = 5 \times 10^{-2}m$	$h = 5 \times 10^{-3}m$	$h = 5 \times 10^{-4}m$
1.0	5.28	6.08	5.45	5.16
0.7	5.24	6.01	5.41	5.11
0.4	5.14	6.10	5.32	5.05
0.1	5.60	5.62	5.42	5.13

(b)

AMA Total Iterations				
$\alpha_{width}$	Distance from Dielectric Interface, $h$ , meters			
	$h = 5 \times 10^{-1}m$	$h = 5 \times 10^{-2}m$	$h = 5 \times 10^{-3}m$	$h = 5 \times 10^{-4}m$
1.0	12	16	28	32
0.7	12	16	39	65
0.4	14	19	46	70
0.1	18	30	54	77

(c)

PAMA Total Iterations				
$\alpha_{width}$	Distance from Dielectric Interface, $h$ , meters			
	$h = 5 \times 10^{-1}m$	$h = 5 \times 10^{-2}m$	$h = 5 \times 10^{-3}m$	$h = 5 \times 10^{-4}m$
1.0	8	8	6	4
0.7	8	8	6	4
0.4	9	8	6	4
0.1	10	8	6	4

(d)

Table 4-15: The DEC 3000AXP/500 CPU seconds used by AMA (a) and PAMA (b) to calculate capacitances for the test problems of Figure 4-18, with the corresponding iteration counts (c) and (d).



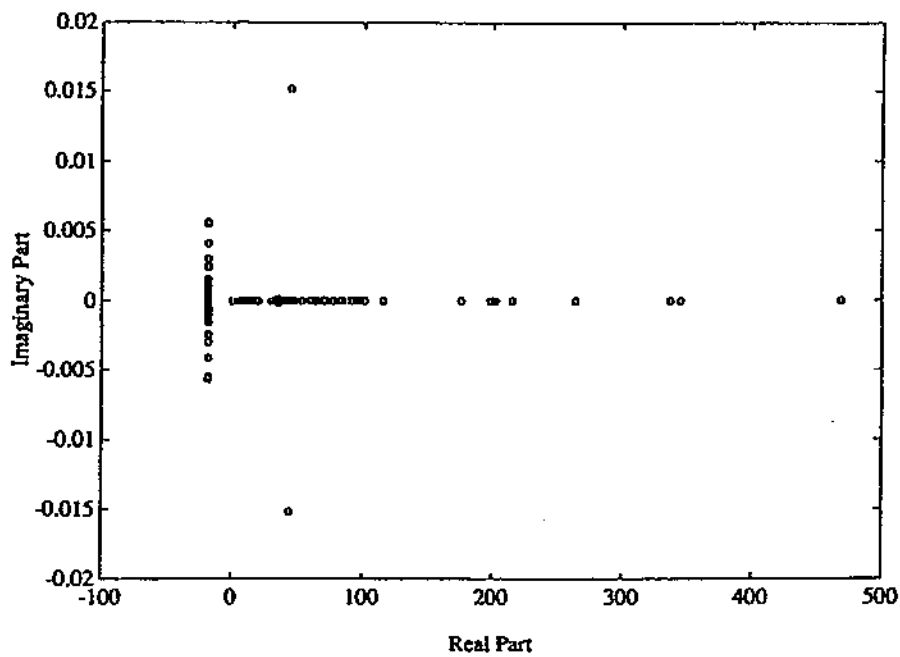
discussed above. The Figure 4-18 geometry is useful for illustrating what sort of discretizations are well conditioned enough to warrant this adjustment of the preconditioner. As Figure 4-18 illustrates, the distance,  $h$ , between the  $1\text{m} \times 1\text{m} \times 1\text{m}$  cubes and the dielectric interface ( $\epsilon_0$  on top,  $2\epsilon_0$  underneath) is varied along with the ratio of the edge-panel widths to the interior-panel widths,  $\alpha_{width}$ , to generate sixteen test problems. The iteration counts and run times for AMA and PAMA applied to the test problems are given in Table 4-15.

The AMA iteration counts in Table 4-15c illustrate how the conditioning worsens as  $h$  and  $\alpha_{width}$  decrease. Figure 4-19a shows the eigenvalues of the effective system matrix used by AMA,  $A + E$  in the terminology of Section 4.2.3, when  $h = 5 \times 10^{-4}$  and  $\alpha_{width} = 0.1$ . The eigenvalues lead to a condition number of  $\kappa(A + E) = 10132$ , indicating that the capacitances are probably in error since AMA uses a residual convergence test (see also Section 3.2.4). In contrast, the PAMA iteration counts are low, and actually decrease as the cubes become closer together, making the nearby panel interactions included in the preconditioner more important. For the  $h = 5 \times 10^{-4}$  and  $\alpha_{width} = 0.1$  case, the preconditioner  $\tilde{C}$  produces the eigenvalue distribution of Figure 4-19b for  $\tilde{C}(A + E)$ , and leads to condition number  $\kappa(\tilde{C}(A + E)) = 30.235$ . Under these circumstances the error bound (4.13) is fairly tight, indicating low error in the computed capacitances.

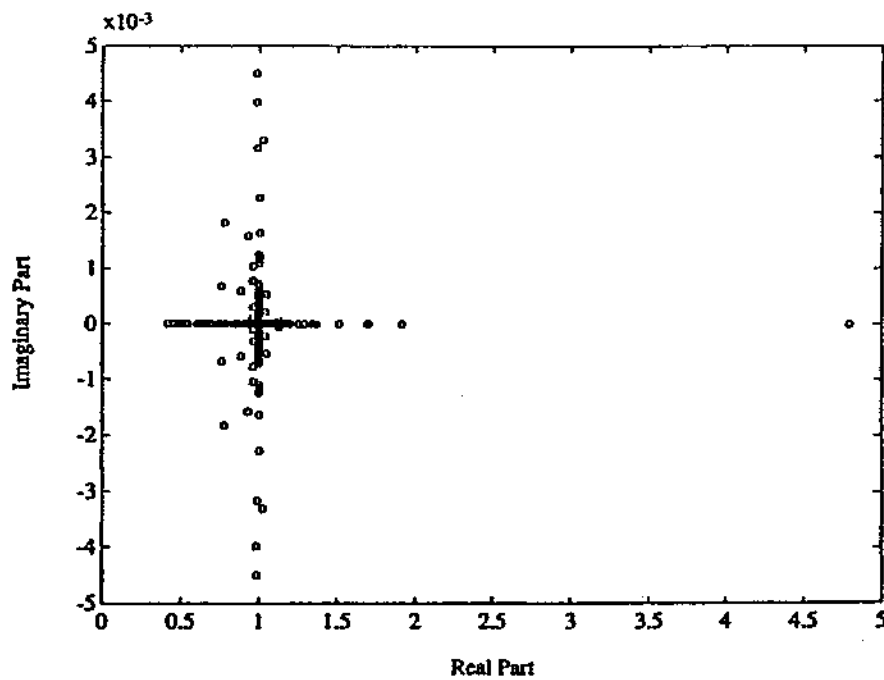
The relatively constant iteration counts for the PAMA method lead to the fairly constant execution times shown in Table 4-15a, while AMA requires increasing time as the conditioning of the problem worsens. For the well conditioned problems with near uniform panel size and well separated surfaces, the PAMA and AMA times become comparable, since the reduction in iteration count afforded by preconditioning is offset by the cost of constructing and applying the preconditioner. In those cases a heuristic based on panel-size variation and panel proximity could be used to determine how much of the nearby panel interaction information should be used to construct the preconditioner. Most practical examples, however, such as the DRAM, diaphragm and connector problems of Table 4-14, require the complete preconditioner used in PAMA.

#### 4.4.3 Memory Use

Figure 4-20 gives plots of the memory required by PAMA to calculate the complete  $4 \times 4$  capacitance matrices for the  $2 \times 2$  woven bus and  $2 \times 2$  coated woven bus as a function of the number of panels  $n$  in the discretizations of Figures 4-9 and 4-10. For these well conditioned problems there is no appreciable difference between the memory used by PAMA and AMA. The same coarsest discretizations for each problem used in the previous section, woven0 and wovendi0, give the smallest-problem results. The four curves in each plot corresponding to the expansion orders  $l = 0, 1, 2$  and  $3$  show that for problems with and without dielectric interfaces the storage complexity is nearly  $O(n)$ , with the expansion order affecting only the

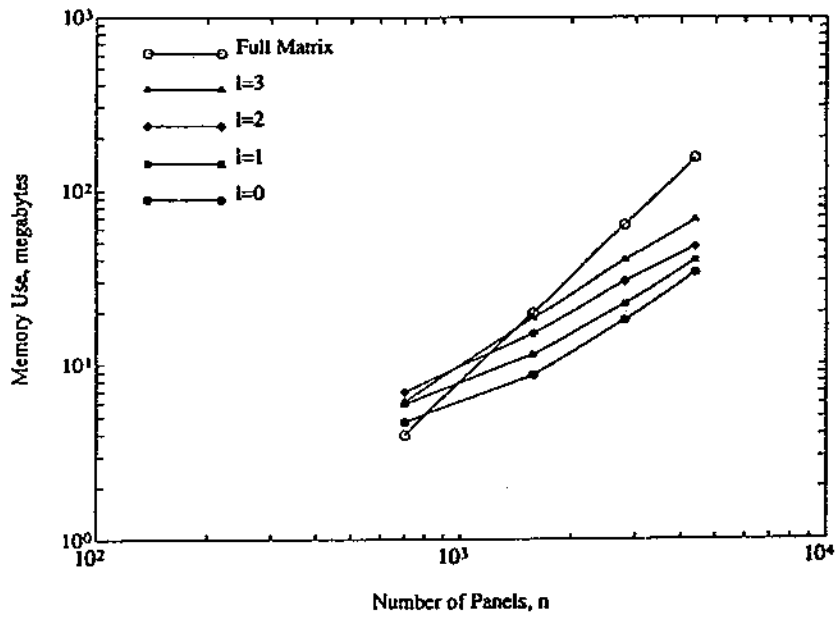


(a)

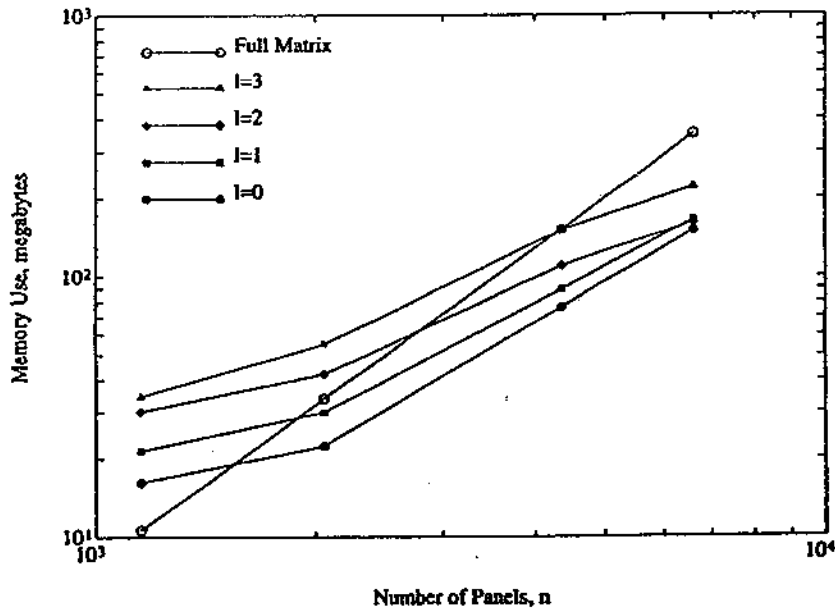


(b)

FIGURE 4-19: Eigenvalues of the effective system matrices used by AMA (a) and PAMA (b) for the Figure 4-18 problem with  $h = 5 \times 10^{-4}$  m and  $\alpha_{width} = 0.1$ .



(a)



(b)

FIGURE 4-20: Memory use in megabytes as a function of number of panels  $n$  for PAMA and the full-matrix methods applied to the woven-bus problem, Figure 4-9, (a) and the coated woven-bus problem, Figure 4-10 (b).

constant that multiplies  $n$  in the expression for the memory<sup>9</sup>. The  $O(n^2)$  memory required to represent the full  $n \times n$  system matrix for both GMRES without multipole acceleration and Gaussian elimination is also plotted. The current implementation uses less memory than the full matrix storage costs for problems with more than a few thousand panels, with somewhat better performance for homogeneous-dielectric problems. Some of the extra memory use is due to the use of the divided-difference algorithm, Algorithm 3.6, rather than the generalized adaptive fast multipole algorithm, Algorithm 3.5, in the current implementation. Algorithm 3.6 requires multiple potential evaluations on each dielectric panel, leading to higher overhead costs.

Homogeneous Dielectric Problems				
Method	Woven2 2816 Panels	Woven3 4400 Panels	Diaphragm 6448 Panels	5 × 5 Woven 9360 Panels
Full Matrix	63.4	155	333	701
MA	45.8	96.5	110	213
AMA	30.0	47.2	51.7	124
PAMA	30.2	47.3	46.1	125
Multiple Dielectric Problems				
Method	Wovendi1 2060 Panels	Wovendi2 4352 Panels	DRAM 6129 Panels	Connector 10096 Panels
Full Matrix	33.9	151	301	815
MA	78.9	280	270	(480)
AMA	42.0	110	120	213
PAMA	42.1	110	117	211

Table 4-16: The memory, in megabytes, required to solve for the capacitance matrices of a representative set of problems. The value in parenthesis is extrapolated.

Table 4-16 summarizes the memory costs associated with the full-matrix and FASTCAP methods applied to the problems of Table 4-14. Generally the FASTCAP memory requirements are smaller than the full-matrix cost, with the difference increasing with  $n$  due to the complexity difference between the two methods. Problems with multiple-dielectric regions, the wovendi1, wovendi2, DRAM and connector problems, use more memory than homogeneous-dielectric problems of similar size due to the previously mentioned overhead of the current implementation. The excessive memory use of the non-adaptive MA method prevents the connector-capacitance calculation, and is evident for all the other problems. In most cases the PAMA memory use is about the same as for AMA since the preconditioner storage is less than 5% of the full matrix cost for problems larger than a few thousand panels, and the reduction in iterations reduces the memory needed to store the  $v^j$  vectors in the iterative loop, Algorithm 4.1. For some poorly conditioned problems, like the DRAM and connector problems, the

<sup>9</sup>Experiments with  $l < 2$  use depth = 4 rather than *auto* to avoid excessive spatial partitioning levels.

savings in iterate storage are enough to cause an appreciable difference between the PAMA and AMA storage requirements.



---

---

## Conclusion and Future Work

The preconditioned multipole accelerated iterative capacitance calculation method studied in this thesis uses nearly  $O(mn)$  computation time and  $O(n)$  storage. The lower complexity results in faster execution times and lower memory use when compared to the standard Gaussian elimination method for problems with more than one- or two-thousand panels. For problems with  $n \approx 10000$ , the new method typically uses hundreds of times less computation time than Gaussian elimination. Calculations that previously required ten hours now require a few minutes, allowing accurate capacitance computations to be part of an iterative design process. Such computations are possible while still maintaining 1% accuracy in the computed capacitances. Execution time can be smoothly exchanged for accuracy if 1% is insufficient, without losing the  $O(mn)$  complexity. The method is completely three-dimensional and has the same complexity and accuracy regardless of the geometry, including problems with multiple dielectric regions. Furthermore, bounds on the algorithm's complexity and error that predict this performance can be derived analytically.

This thesis demonstrates the basic effectiveness of the multipole acceleration technique in capacitance calculations, but many refinements to the algorithm are possible. The standard techniques used to increase the accuracy of the weighted residuals method, such as more complicated expansion functions combined with collocation or Galerkin testing, should increase the accuracy of the method and result in better conditioned matrix problems. Such a refinement may only be useful, however, for the dielectric interface panels, since collocation of constant-charge-density panels appears to be adequate for conductor panels.

Several improvements are possible for the preconditioned multipole acceleration method. The current implementation stops refining the spatial partitioning when at least 90% of the cubes on the lowest level are exact for both the local and multipole expansions (see Sections 3.2.3 and 4.1). This strategy can lead to ineffective preconditioners for well conditioned problems (see Section 4.4.2), although it results in reasonable performance for the adaptive algorithm. A more careful partitioning depth selection method would resolve the conflicting needs of the

preconditioner and the adaptive algorithm. The spatial-partitioning technique also has an effect on the number of panels that straddle cube boundaries. In the worst case a poorly conditioned discretization can have very large panels that extend through many lowest-level cubes, while other much smaller panels fit easily inside the lowest-level cubes. Breaking up the larger panels appears to be a possible solution to this problem. Closely related to this is the idea of adaptively refining panels in order to maintain accuracy. The solution to the adaptive panel refinement problem requires a careful study of the modeling issues involved in order to determine what kind of refinements improve accuracy.

Other improvements involve minor optimizations of the implementation. Currently all panels that are converted to multipole expansions are first approximated as point charges. This approximation is adequate for 1% calculations, but can compromise higher-accuracy computations. Converting the panel charges directly to multipole coefficients requires some modification to the Q2M operator. Part of the reason PAMA performs poorly for well conditioned problems is due to inefficiencies in the realization of the `Delete_Rows()` function called during the preconditioner construction phase of the multipole-algorithm initialization. A more careful treatment of the preconditioner data structures would make `Delete_Rows` more efficient and decrease the preconditioner build time. The implementation would be more accurate and time efficient for multiple-dielectric problems if the divided-difference method, Algorithm 3.6, were replaced with the generalized adaptive multipole algorithm, Algorithm 3.5. The translation (M2M, L2L and M2L) operations could be improved, especially for higher orders, by first rotating the source expansion and then translating it to the destination [113]. Improvements in the representations of the expansions themselves are also possible. The current implementation uses spherical-coordinate expansions and transformation formulas (see Appendix B), which have simple Cartesian equivalents for the low-order expansions useful for capacitance calculations. Converting to the Cartesian representation would eliminate transcendental function evaluations during the initialization phase, making the calculation faster. Finally, the current GMRES iterative loop computes the  $m$  potential problem solutions one at a time. Since all  $m$  problems use the same system matrix with different right-hand-sides, a block GMRES method might lead to savings in run time (see also Section 3.2.1).

Certain geometries have symmetries that could lead to reduced work using the multipole algorithm acceleration. The capacitance of any conductor in the presence of an infinite ground plane can be calculated by the method of images [61]. The problem is equivalent to the original conductor and its image in the ground plane, with the ground plane removed. Because of the symmetry, the two objects give rise to multipole-expansion hierarchies that are rotated and shifted versions of each other. The local expansions differ slightly because of interaction lists that contain cubes on both sides of the image plane, but careful bookkeeping would probably result in some savings. Similar situations occur when calculating capacitances for repeated structures, such as arrays of connector pins.



Extensions of the multipole acceleration technique to other problems such as potential flow [85] and inductance computation [65] have already been made. In general, any problem that can be solved using an iterative method that requires the repeated calculation of potentials or electric fields is a candidate for multipole acceleration. Another such application would be certain discretizations of the diffusion equation, where a potential problem must be solved on every time step. Generalizations of the method to problems that lack the structure of Laplace's equation, but still involve quantities that are less important when farther away, may exploit similarities between the multipole algorithm and multiresolution methods, such as wavelet [10, 62] and multigrid [3] techniques.



---



---

## Surface Layers

This appendix examines the discontinuities in the potential and its normal derivative that occur across charge and dipole surface layers. As described in Sections 2.1.2 and 3.3, charge layers are used to convert the capacitance-calculation potential problem into a simpler, equivalent problem. Charge layers replace conductor surfaces and dielectric interfaces, providing the same discontinuities in the normal derivative of the surface potential. Section 2.1.1 describes similar techniques involving dipole layers<sup>1</sup>. The derivations in this section assume a smooth surface [14, 69, 126, 63]. A careful discussion of how the nature of the surface affects the results, including their extensions for points on conductor edges, is given in [69].

### A.1 Charge Layers

The potential at a point  $x$  due to a single point charge at a different point  $x'$  is, by Coulomb's Law,

$$\psi(x) = \frac{1}{4\pi\epsilon\|x - x'\|_2}, \quad (\text{A.1})$$

assuming an isotropic dielectric of permittivity  $\epsilon$  throughout space. A charge layer density may be thought of as a superposition of infinitesimally small point charges arranged on a surface  $S$  and characterized by a surface-charge density  $\sigma$ , giving rise to the potential

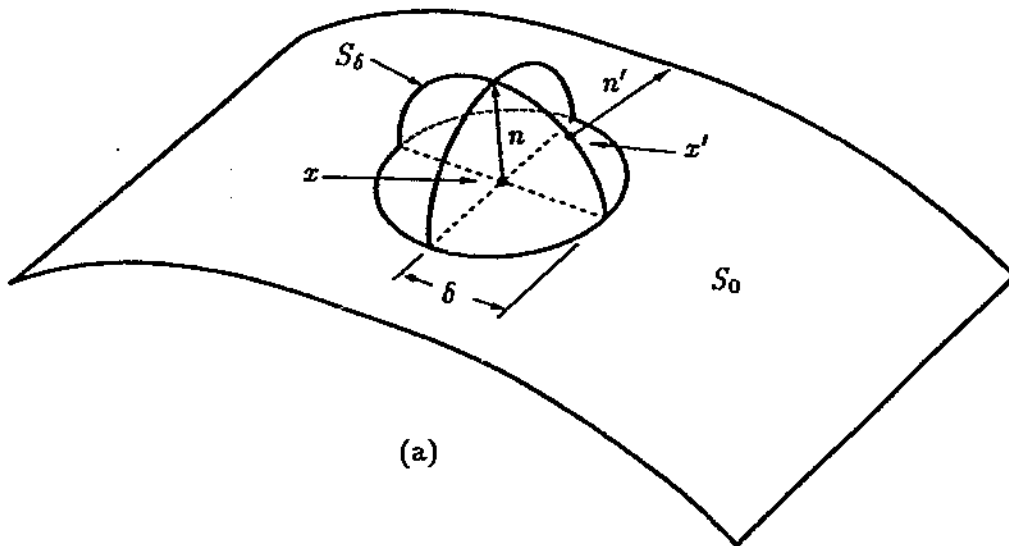
$$\psi(x) = \iint_S \sigma(x') \frac{1}{4\pi\epsilon\|x - x'\|_2} dS', \quad (\text{A.2})$$

where  $x$  cannot lie on  $S$ .

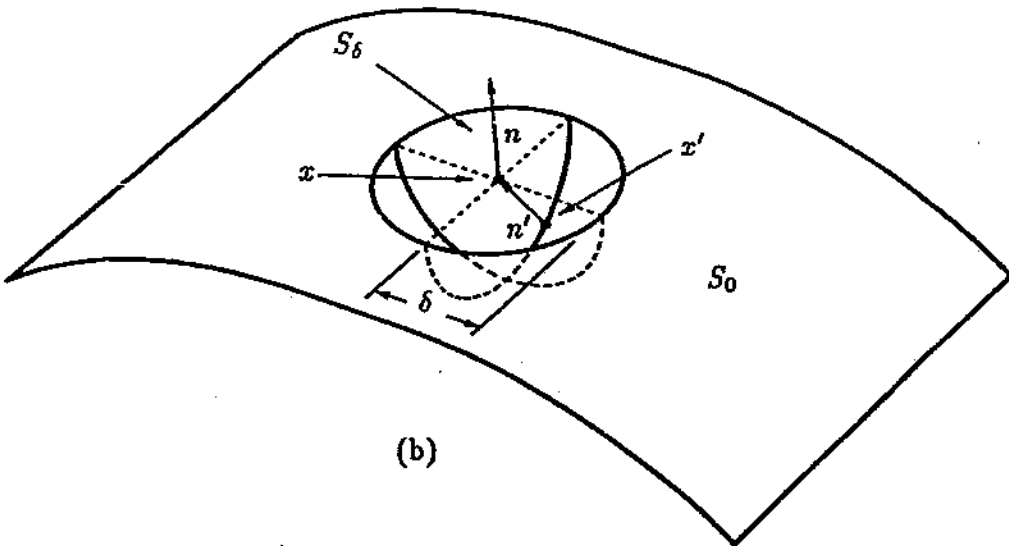
In order to relate the surface potential to the charge density, (A.2) is generalized to allow  $x$  on  $S$  by a limiting argument. In Figure A-1a, as  $\delta$  goes to zero, the domed surface  $S_\delta$  vanishes, leaving  $x$  on the surface  $S$ . Calling the part of  $S$  not replaced by the dome  $S_0$ , the potential at

---

<sup>1</sup>The same formulations result when Green's Theorem [61] is applied to the problems, although the analysis is somewhat less intuitive. See, for example, [4].



(a)



(b)

FIGURE A-1: Taking the limit from the positive (a) and negative (b) sides of the surface.

$x$  can be written as

$$\psi(x) = \lim_{\delta \rightarrow 0} \left[ \iint_{S_\delta} \sigma(x') \frac{1}{4\pi\epsilon \|x - x'\|_2} dS'_\delta + \iint_{S_0} \sigma(x') \frac{1}{4\pi\epsilon \|x - x'\|_2} dS'_0 \right]. \quad (\text{A.3})$$

For  $\delta$  small enough,  $S_\delta$  may be taken as a hemisphere with constant charge density so the first term reduces to

$$\begin{aligned} \lim_{\delta \rightarrow 0} \iint_{S_\delta} \sigma(x') \frac{1}{4\pi\epsilon \|x - x'\|_2} dS'_\delta &= \lim_{\delta \rightarrow 0} \frac{\sigma}{4\pi\epsilon\delta} \iint_{S_\delta} dS'_\delta; \\ &= \lim_{\delta \rightarrow 0} \frac{2\pi\delta^2\sigma}{4\pi\epsilon\delta} = 0. \end{aligned}$$

Thus only the second term in (A.3) contributes, giving a result identical to (A.2) since  $S_0$  tends to  $S$  as  $\delta$  goes to zero. The expression for the potential due to a surface-charge density is always (A.2), even if  $x$  is on the surface<sup>2</sup>. The same result is obtained using the geometry of Figure A-1b and taking the limit from the other side of  $S$ .

Although the potential due to a surface-charge density is continuous across the surface  $S$ , the normal derivative of the potential is not. Calling  $n$  the outward normal to the surface at  $x$ , the derivative of the potential at a point  $s$  relative to  $n$  is calculated by differentiating (A.2),

$$\frac{\partial\psi(s)}{\partial n} = \iint_S \sigma(x') \frac{\partial}{\partial n} \frac{1}{4\pi\epsilon \|s - x'\|_2} dS', \quad (\text{A.4})$$

where  $n$  points into a medium of permittivity  $\epsilon$ . By definition the normal derivative of a function is the component of its gradient in the normal vector's direction,

$$\begin{aligned} \frac{\partial}{\partial n} \frac{1}{4\pi\epsilon \|s - x'\|_2} &\triangleq n \cdot \nabla_s \frac{1}{4\pi\epsilon \|s - x'\|_2}; \\ &= -n \cdot \frac{(s - x')}{4\pi\epsilon \|s - x'\|_2^3}; \\ &= \frac{-\cos(s - x', n)}{4\pi\epsilon \|s - x'\|_2^2}. \end{aligned} \quad (\text{A.5})$$

Here  $\cos(s - x', n)$  is the cosine of the angle between  $s - x'$  and  $n$ ,

$$\cos(s - x', n) \triangleq \frac{n \cdot (s - x')}{\|s - x'\|_2 \|n\|_2}.$$

With this substitution (A.4) becomes

$$\frac{\partial\psi(s)}{\partial n} = - \iint_S \sigma(x') \frac{\cos(s - x', n)}{4\pi\epsilon \|s - x'\|_2^2} dS', \quad (\text{A.6})$$

which is valid for any  $s$  not on the surface  $S$ .

<sup>2</sup>Strictly speaking, the integral does change, from an ordinary integral to a convergent improper integral.

To extend (A.6) to the case of the normal derivative at a point  $x$  on the charge layer surface, a limiting argument based on the geometry of Figure A-1a is used. Using (A.6), the normal derivative of the potential at point  $x$  is

$$\begin{aligned}\frac{\partial\psi_+(x)}{\partial n} &= \lim_{\delta\rightarrow 0} \left[ -\iint_{S_\delta} \sigma(x') \frac{\cos(x-x',n)}{4\pi\epsilon\|x-x'\|_2^2} dS'_\delta - \iint_{S_0} \sigma(x') \frac{\cos(x-x',n)}{4\pi\epsilon\|x-x'\|_2^2} dS'_0 \right]; \\ &= \lim_{\delta\rightarrow 0} \left[ -\frac{\sigma(x)}{4\pi\epsilon\delta^2} \iint_{S_\delta} \cos(x-x',n) dS'_\delta \right] - \iint_S \sigma(x') \frac{\cos(x-x',n)}{4\pi\epsilon\|x-x'\|_2^2} dS'. \quad (\text{A.7})\end{aligned}$$

Using a spherical coordinate system with  $n$  as the  $z$  axis, and  $\theta$  and  $\phi$  as elevation and azimuth coordinates respectively, the integral over the domed section becomes

$$\begin{aligned}\lim_{\delta\rightarrow 0} \left[ -\frac{\sigma(x)}{4\pi\epsilon\delta^2} \iint_{S_\delta} \cos(x-x',n) dS'_\delta \right] &= \lim_{\delta\rightarrow 0} \left[ -\frac{\sigma(x)}{4\pi\epsilon\delta^2} \int_0^{\pi/2} \int_0^{2\pi} \cos\theta \sin\theta \delta^2 d\phi d\theta \right]; \\ &= -\frac{1}{2\epsilon} \sigma(x).\end{aligned}$$

Substitution into (A.7) gives the final expression for the normal derivative of the potential on a charge layer,

$$\frac{\partial\psi_+(x)}{\partial n} = -\frac{1}{2\epsilon} \sigma(x) - \iint_S \sigma(x') \frac{\cos(x-x',n)}{4\pi\epsilon\|x-x'\|_2^2} dS', \quad x \in S. \quad (\text{A.8})$$

By convention  $n$  points away from the positive side of  $S$ . In deriving (A.8), the dome approaches  $S$  from its positive side as in Figure A-1a. Thus (A.8) is valid for the normal derivative of the potential on the positive side of  $S$ . An analogous argument, using the geometry of Figure A-1b, leads to an expression for the normal derivative of the potential on the negative side,

$$\frac{\partial\psi_-(x)}{\partial n} = \frac{1}{2\epsilon} \sigma(x) - \iint_S \sigma(x') \frac{\cos(x-x',n)}{4\pi\epsilon\|x-x'\|_2^2} dS', \quad x \in S. \quad (\text{A.9})$$

Together (A.8) and (A.9) indicate that the normal derivative of the potential has a discontinuity across a surface-charge layer of magnitude

$$\frac{\partial\psi_-(x)}{\partial n} - \frac{\partial\psi_+(x)}{\partial n} = \frac{1}{\epsilon} \sigma(x).$$

Both the integrals in (A.8) and (A.9) must now be interpreted as convergent improper integrals.

## A.2 Dipole Layers

The potential at  $x$  due to a single point charge at  $x'$  in a medium of permittivity  $\epsilon$  is given by (A.1), but may also be defined as the solution to the equation,

$$\nabla^2\psi(x) = -\frac{1}{\epsilon} \delta(x-x'), \quad (\text{A.10})$$

where  $\delta(\cdot)$  is the Dirac delta function. A dipole consists of two unit charges with opposite polarity aligned along a unit vector  $p$ . Specifically, the solution to the equation

$$\nabla^2\psi_d(x) = -\frac{1}{\epsilon} \lim_{h\rightarrow 0} \frac{1}{h} \{ \delta[x - (x' + hp)] - \delta(x-x') \} \quad (\text{A.11})$$

is the potential of a unit dipole at  $x$  with moment  $p$ . By superposition, the two terms in (A.11) must give rise to potentials of the form of (A.1) so

$$\psi_d(x) = \lim_{h \rightarrow 0} \frac{1}{h} \left[ \frac{1}{4\pi\epsilon \|x - (x' + hp)\|_2} - \frac{1}{4\pi\epsilon \|x - x'\|_2} \right].$$

Thus the dipole potential is the  $p$ -directional derivative with respect to  $x'$  of the potential of a point charge at  $x$ ,

$$\psi_d(x) = \frac{\partial}{\partial p} \frac{1}{4\pi\epsilon \|x - x'\|_2}.$$

Using reasoning similar to that leading to (A.5), the potential of a unit dipole at  $x$  oriented in the  $p$  direction is

$$\psi_d(x) = \frac{\cos(x - x', p)}{4\pi\epsilon \|x - x'\|_2^2}.$$

A dipole layer is formed when infinitesimally small dipoles are distributed on a surface  $S$ , oriented in the direction of the surface's normal  $n'$ . Such a distribution is characterized by a dipole layer density  $\mu$  which produces a potential  $\psi_d(x)$  due to the superposition of the infinitesimally small dipoles,

$$\psi_d(x) = \iint_S \mu(x') \frac{\cos(x - x', n')}{4\pi\epsilon \|x - x'\|_2^2} dS'. \quad (\text{A.12})$$

This equation is only valid for  $x$  not on  $S$ .

The extension of (A.12) to points on the dipole layer follows the same strategy as used to develop (A.8) and (A.9). In fact (A.6) and (A.12) differ only in sign and the treatment of the normal, allowing the use of the same limiting argument with slight changes in the dome integration. The expression

$$\psi_{d+}(x) = \frac{1}{2\epsilon} \mu(x) + \iint_S \mu(x') \frac{\cos(x - x', n')}{4\pi\epsilon \|x - x'\|_2^2} dS', \quad x \in S, \quad (\text{A.13})$$

results when the limit is taken from the positive side of  $S$  using the geometry of Figure A-1a. Taking the limit from the negative side of  $S$  as in Figure A-1b gives

$$\psi_{d-}(x) = -\frac{1}{2\epsilon} \mu(x) + \iint_S \mu(x') \frac{\cos(x - x', n')}{4\pi\epsilon \|x - x'\|_2^2} dS', \quad x \in S. \quad (\text{A.14})$$

In (A.13) and (A.14), the integrals are convergent, improper integrals. The potential is discontinuous across a dipole layer since

$$\psi_{d-}(x) - \psi_{d+}(x) = -\frac{1}{\epsilon} \mu(x).$$

The normal derivative of the potential, however, is continuous across a dipole layer. This fact can be established using limiting arguments similar to those leading to the above results.





---



---

## Multipole Algorithm Formulas

This appendix presents the multipole algorithm expansion, shift and conversion formulas used in the capacitance extraction algorithm implementation. The formulas used are equivalent to those in the original multipole algorithm formulation of [52, 54] but avoid complex arithmetic. They are obtained by combining complex conjugates in the original formulas to obtain expressions in the style of [59]. Section B.1 defines the real valued coefficients and the spherical harmonics which together are used to form the multipole formulas of Section B.2. Gaussian units are used throughout this appendix, leading to the simpler formulas used in the implementation. For a comparison of MKSA and Gaussian units see [61].

### B.1 Formula Components

Each multipole or local expansion term involves a coefficient multiplying a spherical harmonic. When a real coefficient expansion is used this fact is obscured by the combination of complex conjugates. However, since the real coefficient expansions are just reorganizations of the complex coefficient formulas, the same coefficients and spherical harmonics appear in slightly different form.

#### B.1.1 Real Valued Expansion Coefficients

Given a multipole or local expansion coefficient,  $G_n^m$ , the corresponding real valued coefficients are defined as

$$\bar{G}_n^m \triangleq \begin{cases} 2\sqrt{\frac{(n+|m|)!}{(n-|m|)!}} \operatorname{Re}\{G_n^m\}, & |m| > 0, |m| \leq n; \\ G_n^m, & m = 0, m \leq n; \\ 0, & \text{otherwise;} \end{cases} \quad (\text{B.1})$$

$$\tilde{G}_n^m \triangleq \begin{cases} -2\sqrt{\frac{(n+m)!}{(n-m)!}} \operatorname{Im}\{G_n^m\}, & m > 0, m \leq n; \\ 2\sqrt{\frac{(n+|m|)!}{(n-|m|)!}} \operatorname{Im}\{G_n^m\}, & m < 0, |m| \leq n; \\ 0, & \text{otherwise.} \end{cases} \quad (\text{B.2})$$

All the multipole algorithm formulas are converted to real coefficients using these substitutions for the complex coefficients.

### B.1.2 Spherical Harmonics

The functions

$$Y_n^m(\theta, \phi) \triangleq \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta) e^{im\phi}, \quad (\text{B.3})$$

are called surface spherical harmonics. A surface spherical harmonic is part of a solution to Laplace's equation obtained by separation of variables. Here, as in [66, 59, 61], a surface spherical harmonic is the product of the elevation ( $\theta$ ) and azimuth ( $\phi$ ) components of the solution. Unlike the usual definition, however, a normalization constant is omitted following [52, 54]. The complete solution is a product of  $Y_n^m$  and a power of  $r$ , the radial coordinate. The result, for example

$$\frac{1}{r^{n+1}} Y_n^m(\theta, \phi), \quad (\text{B.4})$$

is called a spherical harmonic.

The function  $P_n^m(\cos \theta)$  is the associated Legendre function of the first kind with degree  $n$  and order  $m$ . These functions are defined only when  $n$  is a non-negative integer and  $m$  is an integer such that  $-n \leq m \leq n$ . For convenience any  $P_n^m(\cos \theta)$  whose indices do not satisfy these restrictions is taken to be zero.

The recursion

$$\begin{aligned} (n-m)P_n^m(\cos \alpha) \\ = (2n-1) \cos \alpha P_{n-1}^m(\cos \alpha) - (n+m-1)P_{n-2}^m(\cos \alpha), \end{aligned} \quad (\text{B.5})$$

valid for  $0 \leq m \leq n-2$ , and the formulas

$$P_m^m(\cos \alpha) = \frac{(2m)!}{2^m m!} (-\sin \alpha)^m, \quad 0 \leq m, \quad (\text{B.6})$$

$$P_{m+1}^m(\cos \alpha) = (2m+1) \cos \alpha P_m^m(\cos \alpha), \quad 0 \leq m, \quad (\text{B.7})$$

can be used to recursively evaluate the Legendre functions [59, 75].

## B.2 Real Coefficient Multipole Algorithm Formulas

Using the real valued coefficients and the spherical harmonics of the previous section, the multipole algorithm formulas used in the capacitance extraction algorithm are obtained. The

resulting real coefficient formulas eliminate the need for complex arithmetic and square root calculations.

### B.2.1 Multipole Expansion (Q2M, M2P)

The order  $l$  multipole-expansion approximation to the potential,  $\psi$ , at the point  $(r, \theta, \phi)$  is

$$\psi(r, \theta, \phi) \approx \sum_{n=0}^l \sum_{m=-n}^n \frac{M_n^m}{r^{n+1}} Y_n^m(\theta, \phi). \quad (\text{B.8})$$

Applying the definition of the surface spherical harmonic,  $Y_n^m$ , gives

$$\psi(r, \theta, \phi) \approx \sum_{n=0}^l \sum_{m=-n}^n \frac{M_n^m}{r^{n+1}} \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta) e^{im\phi}. \quad (\text{B.9})$$

Substituting the real coefficients using (B.1) and (B.2) yields the real coefficient multipole expansion,

$$\psi(r, \theta, \phi) \approx \sum_{n=0}^l \frac{1}{r^{n+1}} \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) \left[ \bar{M}_n^m \cos(m\phi) + \bar{M}_n^m \sin(m\phi) \right]. \quad (\text{B.10})$$

The complex coefficient local expansion conversion is nearly identical.

A multipole expansion is constructed from  $k$  charges with strengths  $q_i$  and positions  $(\rho_i, \alpha_i, \beta_i)$ ,  $i = 1, \dots, k$ , using

$$M_n^m \triangleq \sum_{i=1}^k q_i \rho_i^n Y_n^{-m}(\alpha_i, \beta_i). \quad (\text{B.11})$$

Substituting (B.1) and (B.2) gives expressions for the real multipole coefficients corresponding to a set of  $k$  charges,

$$\bar{M}_n^m = \begin{cases} 2 \sum_{i=1}^k q_i \rho_i^n P_n^{|m|}(\cos \alpha_i) \cos(m\beta_i), & |m| > 0, |m| \leq n; \\ \sum_{i=1}^k q_i \rho_i^n P_n^0(\cos \alpha_i), & m = 0, m \leq n; \\ 0, & \text{otherwise;} \end{cases} \quad (\text{B.12})$$

$$\bar{M}_n^m = \begin{cases} 2 \sum_{i=1}^k q_i \rho_i^n P_n^{|m|}(\cos \alpha_i) \sin(m\beta_i), & |m| > 0, |m| \leq n; \\ 0, & \text{otherwise.} \end{cases} \quad (\text{B.13})$$

The extension of these equations to charge layers is straightforward.

### B.2.2 Local Expansion (Q2L, L2P)

The order  $l$  local expansion approximation to the potential,  $\psi$ , at the point  $(r, \theta, \phi)$  is

$$\psi(r, \theta, \phi) \approx \sum_{n=0}^l \sum_{m=-n}^n L_n^m Y_n^m(\theta, \phi) r^n, \quad (\text{B.14})$$

or

$$\psi(r, \theta, \phi) \approx \sum_{n=0}^l \sum_{m=-n}^n L_n^m \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta) e^{im\phi} r^n. \quad (\text{B.15})$$

Substituting the real coefficients using (B.1) and (B.2) yields the real coefficient local expansion,

$$\psi(r, \theta, \phi) \approx \sum_{n=0}^l r^n \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) \left[ \bar{L}_n^m \cos(m\phi) + \tilde{L}_n^m \sin(m\phi) \right]. \quad (\text{B.16})$$

A local expansion is constructed from  $k$  charges with strengths  $q_i$  and positions  $(\rho_i, \alpha_i, \beta_i)$ ,  $i = 1, \dots, k$ , using

$$L_n^m \triangleq \sum_{i=1}^k \frac{q_i}{\rho_i^{n+1}} Y_n^{-m}(\alpha_i, \beta_i). \quad (\text{B.17})$$

Substituting (B.1) and (B.2) gives expressions for the real multipole coefficients corresponding to a set of charges,

$$\bar{L}_n^m = \begin{cases} 2 \sum_{i=1}^k \frac{q_i}{\rho_i^{n+1}} P_n^{|m|}(\cos \alpha_i) \cos(m\beta_i), & |m| > 0, |m| \leq n; \\ \sum_{i=1}^k \frac{q_i}{\rho_i^{n+1}} P_n^0(\cos \alpha_i), & m = 0, m \leq n; \\ 0, & \text{otherwise;} \end{cases} \quad (\text{B.18})$$

$$\tilde{L}_n^m = \begin{cases} 2 \sum_{i=1}^k \frac{q_i}{\rho_i^{n+1}} P_n^{|m|}(\cos \alpha_i) \sin(m\beta_i), & |m| > 0, |m| \leq n; \\ 0, & \text{otherwise.} \end{cases} \quad (\text{B.19})$$

The extension of these equations to charge layers is straightforward.

### B.2.3 Multipole-Expansion Shift (M2M)

Consider a multipole expansion about the point  $(\rho, \alpha, \beta)$ . The potential at a given point results when its coordinates relative to  $(\rho, \alpha, \beta)$  are substituted into the expansion. If the expansion about  $(\rho, \alpha, \beta)$  has coefficients  $O_n^m$ , then the coefficients of a shifted multipole expansion about the origin,  $N_j^k$ , are given by

$$N_j^k = \sum_{n=0}^j \sum_{m=-n}^n \frac{\sqrt{(j+k)!(j-k)!} i^{|k|-|m|-|k-m|} Y_n^{-m}(\alpha, \beta) O_{j-n}^{k-m} \rho^n}{\sqrt{(j-n+k-m)!(j-n-k+m)!(n+m)!(n-m)!}}. \quad (\text{B.20})$$

Substituting for the surface spherical harmonics using (B.3) and for the complex coefficients with (B.1) and (B.2) gives the real coefficient multipole-expansion shift formulas for  $j \geq k \geq 0$ ,

$$\begin{aligned} \bar{N}_j^k &= (j+k)! \sum_{n=0}^j \rho^n \sum_{m=0}^n f_M(m, k) \frac{P_n^m(\cos \alpha)}{(n+m)!} \\ &\cdot \left\{ \frac{i^{k-m-|k-m|}}{(j-n+|k-m|)!} \left[ \bar{O}_{j-n}^{k-m} \cos(m\beta) - \tilde{O}_{j-n}^{k-m} \sin(m\beta) \right] \right. \\ &\left. + \frac{(-1)^m}{(j-n+k+m)!} \left[ \bar{O}_{j-n}^{k+m} \cos(m\beta) + \tilde{O}_{j-n}^{k+m} \sin(m\beta) \right] \right\}; \end{aligned} \quad (\text{B.21})$$

$$\begin{aligned} \bar{N}_j^k &= (j+k)! \sum_{n=0}^j \rho^n \sum_{m=0}^n f_M(m, k) \frac{P_n^m(\cos \alpha)}{(n+m)!} \\ &\cdot \left\{ \frac{i^{k-m-|k-m|}}{(j-n+|k-m|)!} \left[ \bar{O}_{j-n}^{k-m} \sin(m\beta) + \bar{O}_{j-n}^{k-m} \cos(m\beta) \right] \right. \\ &\left. + \frac{(-1)^m}{(j-n+k+m)!} \left[ -\bar{O}_{j-n}^{k+m} \sin(m\beta) + \bar{O}_{j-n}^{k+m} \cos(m\beta) \right] \right\}. \end{aligned} \quad (\text{B.22})$$

Here

$$f_M(m, k) \triangleq \begin{cases} 1, & m \neq 0, k \neq 0; \\ 1/2, & m = 0, k \neq 0; \\ 1/2, & m \neq 0, k = 0; \\ 1/2, & m = 0, k = 0. \end{cases} \quad (\text{B.23})$$

#### B.2.4 Multipole to Local Expansion Conversion (M2L)

An order  $l$  multipole expansion about the point  $(\rho, \alpha, \beta)$ , with coefficients  $O_n^m$ , can be converted to an order  $l$  local expansion about the origin, with coefficients  $N_j^k$ , using

$$N_j^k = \sum_{n=0}^l \sum_{m=-n}^n \frac{\sqrt{(j+n+m-k)! (j+n-m+k)!} i^{|k-m|} Y_{j+n}^{m-k}(\alpha, \beta) O_n^m}{\sqrt{(n+m)! (n-m)! (j+k)! (j-k)!} (-1)^n i^{|k|+|m|} \rho^{j+n+1}}. \quad (\text{B.24})$$

Substituting for the surface spherical harmonics using (B.3) and for the complex coefficients with (B.1) and (B.2) gives the real coefficient multipole to local expansion conversion formulas for  $l \geq j \geq k \geq 0$ ,

$$\begin{aligned} \bar{N}_j^k &= \frac{f_L(k)}{\rho^j (j-k)!} \sum_{n=0}^l \frac{(-1)^n}{\rho^{n+1}} \sum_{m=0}^n \\ &P_{j+n}^{|m-k|}(\cos \alpha) \frac{(j+n-|m-k|)!}{(n+m)!} i^{|k-m|-k-m} \\ &\cdot \left\{ \bar{O}_n^m \cos[(m-k)\beta] + \bar{O}_n^m \sin[(m-k)\beta] \right\} \\ &+ P_{j+n}^{m+k}(\cos \alpha) \frac{(j+n-m-k)!}{(n+m)!} \\ &\cdot \left\{ \bar{O}_n^m \cos[(m+k)\beta] + \bar{O}_n^m \sin[(m+k)\beta] \right\}; \end{aligned} \quad (\text{B.25})$$

$$\begin{aligned} \bar{N}_j^k &= \frac{1}{\rho^j (j-k)!} \sum_{n=0}^l \frac{(-1)^n}{\rho^{n+1}} \sum_{m=0}^n \\ &P_{j+n}^{|m-k|}(\cos \alpha) \frac{(j+n-|m-k|)!}{(n+m)!} i^{|k-m|-k-m} \\ &\cdot \left\{ -\bar{O}_n^m \sin[(m-k)\beta] + \bar{O}_n^m \cos[(m-k)\beta] \right\} \\ &+ P_{j+n}^{m+k}(\cos \alpha) \frac{(j+n-m-k)!}{(n+m)!} \\ &\cdot \left\{ \bar{O}_n^m \sin[(m+k)\beta] - \bar{O}_n^m \cos[(m+k)\beta] \right\}. \end{aligned} \quad (\text{B.26})$$

Here

$$f_L(k) = \begin{cases} 1, & k \neq 0; \\ 1/2, & k = 0. \end{cases} \quad (\text{B.27})$$

### B.2.5 Local Expansion Shift (L2L)

An order  $l$  local expansion about the point  $(\rho, \alpha, \beta)$ , with coefficients  $O_n^m$ , can be converted to an order  $l$  local expansion about the origin, with coefficients  $N_j^k$ , using

$$N_j^k = \sum_{n=j}^l \sum_{m=-n}^n \frac{\sqrt{(n+m)!(n-m)!} i^{|m|-|k|-|m-k|} Y_{n-j}^{m-k}(\alpha, \beta) O_n^m \rho^{n-j}}{\sqrt{(n-j+m-k)!(n-j-m+k)!(j+k)!(j-k)!} (-1)^{n-j}}. \quad (\text{B.28})$$

Substituting for the surface spherical harmonics using (B.3) and for the complex coefficients with (B.1) and (B.2) gives the real coefficient local expansion shift formulas for  $j \geq k \geq 0$ ,

$$\begin{aligned} \bar{N}_j^k &= \frac{f_L(k)}{(-\rho)^j (j-k)!} \sum_{n=j}^l (-\rho)^n \sum_{m=0}^n \\ &\quad \frac{i^{m-k-|m-k|}}{(n-j+|m-k|)!(n-m)!} \\ &\quad P_{n-j}^{|m-k|}(\cos \alpha) \cdot \left\{ \bar{O}_n^m \cos[(m-k)\beta] + \bar{O}_n^m \sin[(m-k)\beta] \right\} \\ &\quad + P_{n-j}^{m+k}(\cos \alpha) \frac{(-1)^k}{(n-j+m+k)!(n-m)!} \\ &\quad \cdot \left\{ \bar{O}_n^m \cos[(m+k)\beta] + \bar{O}_n^m \sin[(m+k)\beta] \right\}; \end{aligned} \quad (\text{B.29})$$

$$\begin{aligned} \tilde{N}_j^k &= \frac{1}{(-\rho)^j (j-k)!} \sum_{n=j}^l (-\rho)^n \sum_{m=0}^n \\ &\quad \frac{i^{m-k-|m-k|}}{(n-j+|m-k|)!(n-m)!} \\ &\quad P_{n-j}^{|m-k|}(\cos \alpha) \cdot \left\{ -\bar{O}_n^m \sin[(m-k)\beta] + \bar{O}_n^m \cos[(m-k)\beta] \right\} \\ &\quad + P_{n-j}^{m+k}(\cos \alpha) \frac{(-1)^k}{(n-j+m+k)!(n-m)!} \\ &\quad \cdot \left\{ \bar{O}_n^m \sin[(m+k)\beta] - \bar{O}_n^m \cos[(m+k)\beta] \right\}. \end{aligned} \quad (\text{B.30})$$

The function  $f_L(k)$  is given by (B.27).

### B.2.6 Electric Fields From Multipole Expansions (M2E)

The contribution of an order  $l$  multipole expansion to the potential at the point  $(r, \theta, \phi)$  is given by (B.10). If  $(r, \theta, \phi)$  are the spherical coordinates of a dielectric-panel evaluation point with unit Cartesian-coordinate normal  $\mathbf{n} = (n_x, n_y, n_z)$ , then the multipole expansion (B.10) contributes

$$E_{mut} \triangleq \mathbf{n} \cdot \nabla \left\{ \sum_{n=0}^l \frac{1}{r^{n+1}} \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) \left[ \bar{M}_n^m \cos(m\phi) + \bar{M}_n^m \sin(m\phi) \right] \right\} \quad (\text{B.31})$$

to the electric field in the normal direction. Expanding the gradient in Cartesian coordinates using the chain rule leads to the directional derivatives

$$\frac{\partial}{\partial x} p_{mul}(r, \theta, \phi) = \frac{\partial p_{mul}}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial p_{mul}}{\partial \theta} \frac{\partial \theta}{\partial x} + \frac{\partial p_{mul}}{\partial \phi} \frac{\partial \phi}{\partial x}; \quad (\text{B.32})$$

$$\frac{\partial}{\partial y} p_{mul}(r, \theta, \phi) = \frac{\partial p_{mul}}{\partial r} \frac{\partial r}{\partial y} + \frac{\partial p_{mul}}{\partial \theta} \frac{\partial \theta}{\partial y} + \frac{\partial p_{mul}}{\partial \phi} \frac{\partial \phi}{\partial y}; \quad (\text{B.33})$$

$$\frac{\partial}{\partial z} p_{mul}(r, \theta, \phi) = \frac{\partial p_{mul}}{\partial r} \frac{\partial r}{\partial z} + \frac{\partial p_{mul}}{\partial \theta} \frac{\partial \theta}{\partial z} + \frac{\partial p_{mul}}{\partial \phi} \frac{\partial \phi}{\partial z}. \quad (\text{B.34})$$

Performing the dot product and rearranging gives

$$\begin{aligned} E_{mul} = & \frac{\partial p_{mul}}{\partial r} \left( \frac{\partial r}{\partial x} n_x + \frac{\partial r}{\partial y} n_y + \frac{\partial r}{\partial z} n_z \right) \\ & + \frac{\partial p_{mul}}{\partial \theta} \left( \frac{\partial \theta}{\partial x} n_x + \frac{\partial \theta}{\partial y} n_y + \frac{\partial \theta}{\partial z} n_z \right) \\ & + \frac{\partial p_{mul}}{\partial \phi} \left( \frac{\partial \phi}{\partial x} n_x + \frac{\partial \phi}{\partial y} n_y + \frac{\partial \phi}{\partial z} n_z \right). \end{aligned} \quad (\text{B.35})$$

For the case  $\sin \theta \neq 0$ , derivatives obtained by implicit differentiation of the coordinate conversion rules

$$z = r \cos \theta; \quad x = r \cos \phi \sin \theta; \quad y = r \sin \phi \sin \theta, \quad (\text{B.36})$$

together with various derivatives of (B.10),

$$\frac{\partial p_{mul}}{\partial r} = \sum_{n=0}^l \frac{-(n+1)}{r^{n+2}} \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) \left[ \bar{M}_n^m \cos(m\phi) + \bar{M}_n^m \sin(m\phi) \right]; \quad (\text{B.37})$$

$$\begin{aligned} \frac{\partial p_{mul}}{\partial \theta} = & \sum_{n=0}^l \frac{1}{r^{n+1}} \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} \left[ \frac{(n-m+1)}{\sin \theta} P_{n+1}^m(\cos \theta) - \frac{(n+1) \cos \theta}{\sin \theta} P_n^m(\cos \theta) \right] \\ & \cdot \left[ \bar{M}_n^m \cos(m\phi) + \bar{M}_n^m \sin(m\phi) \right]; \end{aligned} \quad (\text{B.38})$$

$$\frac{\partial p_{mul}}{\partial \phi} = \sum_{n=0}^l \frac{1}{r^{n+1}} \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) \left[ -m \bar{M}_n^m \sin(m\phi) + m \bar{M}_n^m \cos(m\phi) \right], \quad (\text{B.39})$$

$$(\text{B.40})$$

are substituted into (B.35). The derivative of the associated Legendre function in (B.38) has been replaced using the identity [75]

$$\sin \theta \frac{d}{d\theta} P_n^m(\cos \theta) = (n-m+1) P_{n+1}^m(\cos \theta) - (n+1) \cos \theta P_n^m(\cos \theta). \quad (\text{B.41})$$

After the substitutions (B.35) becomes

$$\begin{aligned} E_{mul} = & \sum_{n=0}^l \frac{1}{r^{n+2}} \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} \left\{ \right. \\ & \left. \left\{ A_n \cos(m\phi) P_n^m(\cos \theta) + [F_n^m \cos(m\phi) + B^m \sin(m\phi)] P_{n+1}^m(\cos \theta) \right\} \bar{M}_n^m \right. \\ & + \left\{ [G_n \sin(m\phi) - B^m \cos(m\phi)] P_n^m(\cos \theta) \right. \\ & \left. \left. + F_n^m \sin(m\phi) P_{n+1}^m(\cos \theta) \right\} \bar{M}_n^m \right\}, \quad \sin \theta \neq 0. \end{aligned} \quad (\text{B.42})$$

where

$$A_n \triangleq (n+1) \left[ \frac{\cos \theta}{\sin \theta} (n_x \cos \phi \cos \theta + n_y \sin \phi \cos \theta - n_z \sin \theta) - (n_x \sin \theta \cos \phi + n_y \sin \theta \sin \phi + n_z \cos \theta) \right]; \quad (\text{B.43})$$

$$B^m \triangleq (n_x \sin \phi - n_y \cos \phi) \frac{m}{\sin \theta}; \quad (\text{B.44})$$

$$F_n^m \triangleq \frac{(m-n-1)}{\sin \theta} (n_x \cos \phi \cos \theta + n_y \sin \phi \cos \theta - n_z \sin \theta);$$

$$G_n \triangleq (n+1) \left[ \frac{\cos \theta}{\sin \theta} (n_x \cos \phi \cos \theta + n_y \sin \phi \cos \theta - n_z \sin \theta) - (n_x \sin \theta \cos \phi + n_y \sin \theta \sin \phi + n_z \cos \theta) \right]. \quad (\text{B.45})$$

For the special case  $\sin \theta = 0$  (B.42) is replaced by

$$E_{mul} = \sum_{n=0}^l \frac{-(n+1)}{r^{n+2}} P_n^0(\cos \theta) \bar{M}_n^0, \quad \sin \theta = 0. \quad (\text{B.46})$$

This equation's derivation starts from (B.10) for the special case  $\sin \theta = 0$  and then proceeds as when  $\sin \theta \neq 0$ , with special attention to simplifications in  $P_n^m(\cos \theta)$  and  $\frac{d}{d\theta} P_n^m(\cos \theta)$ .

The normal electric field evaluation at  $x_i = (r, \theta, \phi)$  uses either (B.42) or (B.46), as appropriate, to replace the  $E_{ij}q_j$  products in (3.50), or  $E_{ij}^G q_j$  products in (3.53), corresponding to the charge panels represented by the multipole expansion.

### B.2.7 Electric Fields From Local Expansions (L2E)

The contribution of an order  $l$  local expansion for the potential at the point  $(r, \theta, \phi)$  is

$$p_{loc}(r, \theta, \phi) = \sum_{n=0}^l r^n \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) \left[ \bar{L}_n^m \cos(m\phi) + \bar{L}_n^m \sin(m\phi) \right]. \quad (\text{B.47})$$

The multipole and local expansions differ only in the powers of  $r$  that multiply each term. This similarity allows a development analogous to that in the previous section which gives

$$E_{loc} = \sum_{n=0}^l n r^{n-1} \sum_{m=0}^n \frac{(n-m)!}{(n+m)!} \left\{ \begin{aligned} & \{ A_n \cos(m\phi) P_n^m(\cos \theta) + [F_n^m \cos(m\phi) + B^m \sin(m\phi)] P_{n+1}^m(\cos \theta) \} \bar{L}_n^m \\ & + \{ [G_n \sin(m\phi) - B^m \cos(m\phi)] P_n^m(\cos \theta) \\ & + F_n^m \sin(m\phi) P_{n+1}^m(\cos \theta) \} \bar{L}_n^m \}, \quad \sin \theta \neq 0. \end{aligned} \right. \quad (\text{B.48})$$

for the normal component of the electric field due to the local expansion (B.47) evaluated at dielectric panel center  $(r, \theta, \phi)$ . The weights  $A_n$ ,  $B^m$ ,  $F_n^m$  and  $G_n$  are given by (B.43–B.45) as in the multipole-expansion case. When  $\sin \theta = 0$  the normal field is

$$E_{loc} = \sum_{n=0}^l n r^{n-1} P_n^0(\cos \theta) \bar{M}_n^0, \quad \sin \theta = 0. \quad (\text{B.49})$$



As in the multipole case, the evaluation of the normal electric field at  $x_i = (r, \theta, \phi)$  uses either (B.48) or (B.49), as appropriate, to replace the  $E_{ij}q_j$  products in (3.50), or  $E_{ij}^G q_j$  products in (3.53), corresponding to the charge panels represented by the local expansion.



## Galerkin Panel Integral Approximation

Section 3.3.2 applies a constant-panel-charge approximation and Galerkin testing to the integral equation arising from the boundary condition at the interface between two dielectric regions to obtain (3.52),

$$0 = q_i \frac{(\epsilon_a + \epsilon_b)}{2A_i \epsilon_0 (\epsilon_b - \epsilon_a)} + \sum_{j=1}^n q_j \frac{1}{A_i A_j} \iint_{\text{panel } i} \iint_{\text{panel } j} \frac{\cos(x - x', n_b)}{4\pi \epsilon_0 \|x - x'\|_2^2} dS' dS, \quad x \in S^{ab}, \quad (\text{C.1})$$

where  $S^{ab}$  is the dielectric interface,  $\epsilon_a$  and  $\epsilon_b$  are the dielectric permittivities on either side of the interface,  $n_b$  is the normal to  $S^{ab}$  at  $x$  that points into the  $\epsilon_b$  region,  $S$  is all the conductor and dielectric-interface surfaces in the problem, and  $q_i$  and  $A_i$  are the total charge and area of panel  $i$ . In the collocation-Galerkin discretization of Section 3.3.2, each dielectric panel  $i$  gives rise to an equation of this form, which leads to one row of the matrix  $E^G$  given by (3.54). Solving the corresponding matrix problem (3.53) iteratively requires multiplication of the charge vector  $q$  by  $E^G$  (using Algorithm 3.5 or 3.6), which amounts to evaluating the right-hand side of equations like (C.1) for all the dielectric panels. The second term in these evaluations can be rearranged to read

$$\frac{1}{A_i} \iint_{\text{panel } i} \sum_{j=1}^n \frac{q_j}{A_j} \iint_{\text{panel } j} \frac{\cos(x - x', n_b)}{4\pi \epsilon_0 \|x - x'\|_2^2} dS' dS, \quad x \in S^{ab}. \quad (\text{C.2})$$

By Coulomb's Law the inner integrand is the normal component of the electric field at the point  $x$  on panel  $i$  due to the charge at  $x'$  on panel  $j$ . The panel- $j$  integral adds the contributions of all the charge on panel  $j$ , and the summation extends the superposition to all the panels in the problem. Thus

$$E_{n_b}(x) \triangleq \sum_{j=1}^n \frac{q_j}{A_j} \iint_{\text{panel } j} \frac{\cos(x - x', n_b)}{4\pi \epsilon_0 \|x - x'\|_2^2} dS' dS, \quad x \in S^{ab} \quad (\text{C.3})$$

is the normal electric field at the point  $x$  on panel  $i$  due to all the charge in the problem, except for the charge on panel  $i$ , which contributes through the first term in (C.1).

The expression (C.2) is therefore the integral of the normal electric field over panel  $i$  divided by  $A_i$ , that is the average normal field on panel  $i$ . Since closed form expressions for (C.2) are not known for arbitrary panel orientations, and (C.3) can be approximated with the multipole algorithm, the approach taken here is to approximate the averaging integral numerically using a weighted sum of several evaluations of the integrand (C.3). In Algorithm 3.5 the normal-field evaluations are calculated directly using a generalization of the multipole algorithm, while in Algorithm 3.6 they are approximated from multipole-algorithm potential evaluations using divided differences.

The integral approximation formulas depend on the shape and orientation of the panel. If panel  $i$  is a triangle, the integral

$$\frac{1}{A_i} \iint_{\text{panel } i} E_{n_b}(x) dS \quad (\text{C.4})$$

is first transformed to an integral over the standard triangle with corners  $(0, 0)$ ,  $(1, 0)$  and  $(0, 1)$ . The affine transformation that maps the standard triangle onto panel  $i$  is easily calculated from panel  $i$ 's corner points and has the form

$$x = T\hat{x} + v, \quad (\text{C.5})$$

where  $x$  is any point on panel  $i$ ,  $\hat{x}$  is the corresponding point on the standard triangle,  $T$  is a  $2 \times 2$  matrix and  $v$  is a vector. Using (C.5) to change variables in (C.4) gives [76]

$$\frac{1}{A_i} \iint_{\text{panel } i} E_{n_b}(x) dS = \frac{1}{A_i} \iint_{\text{std. tri.}} E_{n_b}(T\hat{x} + v) |\det(T)| d\hat{S},$$

so that any triangular-panel integral reduces to an integral over the standard triangle. Here  $\det(T)$  is the determinate of  $T$ .

The transformed integral is approximated using an integration rule of the form

$$\frac{1}{A_i} \iint_{\text{std. tri.}} E_{n_b}(T\hat{x} + v) |\det(T)| d\hat{S} \approx |\det(T)| \sum_{k=1}^K E_{n_b}(T\hat{x}_k + v) d_k \quad (\text{C.6})$$

where the evaluation points  $\hat{x}_k$  and weights  $d_k$  are given in Table C-1 for the rules used in the current implementation, Algorithm 4.1. The order  $o = 7$  formula is a Gaussian quadrature formula applied to the standard triangle by a change of variables, while the others are derived using special techniques [128].

A formula's order,  $o$ , is a measure of its accuracy. An order  $o$  formula produces the same result as the actual integral when the integrand is any monomial of order  $p$ ,  $0 \leq p \leq o$ . The formulas of Table C-1 are particularly useful for (C.6) because all their evaluation points are inside the panel, all their weights are positive, and they use a minimal number of evaluation points. Formulas that use evaluation points on the edges of the panel produce inaccurate results

Order, $o$	Evaluation Points, $\hat{x}_k$ , and Weights, $d_k$	Constants
1	$\hat{x}_1 = (1/3, 1/3), d_1 = 1/2$	
2	$\hat{x}_1 = (1/6, 1/6), d_1 = 1/6$ $\hat{x}_2 = (2/3, 1/6), d_2 = 1/6$ $\hat{x}_3 = (1/6, 2/3), d_3 = 1/6$	
3	$\hat{x}_1 = (r, s), d_1 = 1/12$ $\hat{x}_2 = (s, r), d_2 = 1/12$ $\hat{x}_3 = (t, r), d_3 = 1/12$ $\hat{x}_4 = (r, t), d_4 = 1/12$ $\hat{x}_5 = (t, s), d_5 = 1/12$ $\hat{x}_6 = (s, t), d_6 = 1/12$	$r = 0.1090390091$ $s = 0.2319333686$ $t = 0.6590276224$
5	$\hat{x}_1 = (1/3, 1/3), d_1 = 9/80$ $\hat{x}_2 = (r, r), d_2 = B$ $\hat{x}_3 = (r, s), d_3 = B$ $\hat{x}_4 = (s, r), d_4 = B$ $\hat{x}_5 = (u, u), d_5 = C$ $\hat{x}_6 = (u, v), d_6 = C$ $\hat{x}_7 = (v, u), d_7 = C$	$r = (6 - \sqrt{15})/21$ $s = (9 + 2\sqrt{15})/21$ $u = (6 + \sqrt{15})/21$ $v = (9 - 2\sqrt{15})/21$ $B = (155 - \sqrt{15})/2400$ $C = (155 + \sqrt{15})/2400$
7	$\hat{x}_1 = (s_1, r_1(1 - s_1)), d_1 = B_1C_1$ $\hat{x}_2 = (s_1, r_2(1 - s_1)), d_2 = B_2C_1$ $\hat{x}_3 = (s_1, r_3(1 - s_1)), d_3 = B_3C_1$ $\hat{x}_4 = (s_1, r_4(1 - s_1)), d_4 = B_4C_1$ $\hat{x}_5 = (s_2, r_1(1 - s_2)), d_5 = B_1C_2$ $\hat{x}_6 = (s_2, r_2(1 - s_2)), d_6 = B_2C_2$ $\hat{x}_7 = (s_2, r_3(1 - s_2)), d_7 = B_3C_2$ $\hat{x}_8 = (s_2, r_4(1 - s_2)), d_8 = B_4C_2$ $\hat{x}_9 = (s_3, r_1(1 - s_3)), d_9 = B_1C_3$ $\hat{x}_{10} = (s_3, r_2(1 - s_3)), d_{10} = B_2C_3$ $\hat{x}_{11} = (s_3, r_3(1 - s_3)), d_{11} = B_3C_3$ $\hat{x}_{12} = (s_3, r_4(1 - s_3)), d_{12} = B_4C_3$ $\hat{x}_{13} = (s_4, r_1(1 - s_4)), d_{13} = B_1C_4$ $\hat{x}_{14} = (s_4, r_2(1 - s_4)), d_{14} = B_2C_4$ $\hat{x}_{15} = (s_4, r_3(1 - s_4)), d_{15} = B_3C_4$ $\hat{x}_{16} = (s_4, r_4(1 - s_4)), d_{16} = B_4C_4$	$r_1 = 0.0694318422$ $r_2 = 0.3300094782$ $r_3 = 0.6699905218$ $r_4 = 0.9305681558$ $s_1 = 0.0571041961$ $s_2 = 0.2768430136$ $s_3 = 0.5835904324$ $s_4 = 0.8602401357$ $B_1 = 0.1739274226$ $B_2 = 0.3260725774$ $B_3 = 0.3260725774$ $B_4 = 0.1739274226$ $C_1 = 0.1355069134$ $C_2 = 0.2034645680$ $C_3 = 0.1298475476$ $C_4 = 0.0311809709$

Table C-1: Evaluation points and weights used in the approximation of (C.6).

because the panel-charge densities are constant, while formulas with negative weights are less accurate due to cancellation errors. Although quadrilateral panel formulas exist [33, 128], the current implementation treats quadrilateral panel integrals as two triangle integrals, unless  $o = 1$  where centroid collocation is used.

---



---

## Local Expansion Efficiency

Section 3.2.3 establishes that it is not computationally efficient to construct an order- $l$  local expansion for a cube containing  $n_p$  evaluation points if

$$n_p \leq (l + 1)^2. \quad (\text{D.1})$$

If (D.1) is violated for a given cube, however, it is still possible that constructing a local expansion for the given cube and then evaluating it requires more work than directly evaluating the expansions used to build the given cube's expansion. The final cost of the local expansion calculation may be greater since the local expansion requires evaluation, possibly with a hierarchy of lower-level local expansions supporting the eventual evaluation of lowest-level expansions, while the direct evaluation gives the evaluation-point potentials without further computation. The potential savings, however, are not enough to warrant checking for this possibility.

For a given cube that violates (D.1), the cost of direct evaluation must be compared to the cost of building not only the given cube's local-expansion, but also the cost of any local expansions required in its descendent cubes to evaluate the potential at the given cube's evaluation points. The direct evaluation starts by evaluating the parent's local expansion (if present) at all the given cube's evaluation points (L2P), and then adds in all the interaction-list contributions for the given cube and all its descendants (M2P and Q2P). The interaction-list computations have the form of (3.17) without the L2P transformation, with each descendant of the given cube contributing a separate set of  $n_p^j \times n_i^j$  transformation matrices, where the  $j$ -th descendant contains  $n_p^j$  evaluation points and the given cube is considered to be one of its own descendants. The cost of the direct evaluation of all the interaction-list entries is then

$$\sum_{j=1}^J n_p^j \sum_{i=1}^{I_j} n_i^j, \quad (\text{D.2})$$

where  $J$  is the total number of descendent cubes and  $I_j$  is the number of entries in the  $j$ -th descendant's interaction list. If the given cube's parent has no local expansion, then (D.2) is

the total cost of the direct-evaluation strategy, otherwise adding the cost of  $n_p$  evaluations of the  $(l+1)^2$ -term parent-cube local expansion at the given cube's evaluation points gives the total cost,

$$\sum_{j=1}^J n_p^j \sum_{i=1}^{I_j} n_i^j + n_p(l+1)^2. \quad (\text{D.3})$$

The local-expansion computation starts with a translated version of the parent's local expansion (L2L, if present) and adds in the contributions of its interaction-list entries (M2L and Q2L). Assuming that none of the given cube's descendants satisfy (D.1), the same procedure performed recursively constructs a hierarchy of lower-level local expansions for all the descendants of the given cube. Finally the lowest-level local expansions are evaluated (L2P). The interaction-list computation has the form of (3.16) without the the L2L transformation, with each descendant of the given cube contributing a separate set of  $(l+1)^2 \times n_i^j$  transformation matrices towards that descendant's local-expansion build. Each non-empty descendent cube leads to an  $(l+1)^2 \times (l+1)^2$  L2L transformation matrix, assuming the parent's local expansion is present, and the final L2P evaluations require  $n_p$  evaluations of  $(l+1)^2$ -term local expansions, making the total local-expansion computation cost

$$\sum_{j=1}^J (l+1)^2 \sum_{i=1}^{I_j} n_i^j + (J-1)(l+1)^4 + n_p(l+1)^2. \quad (\text{D.4})$$

When the given cube's parent has no local expansion, the computation is identical except that there are no L2L transformations from the parent's local to the given cube's  $\beta$  descendants one level finer than the given cube, so that (D.4) becomes

$$\sum_{j=1}^J (l+1)^2 \sum_{i=1}^{I_j} n_i^j + (J-\beta-1)(l+1)^4 + n_p(l+1)^2. \quad (\text{D.5})$$

Combining (D.2) with (D.5), the direct evaluation of the potentials in a given cube is preferable to a local-expansion computation and evaluation if

$$\sum_{j=1}^J n_p^j \sum_{i=1}^{I_j} n_i^j \leq \sum_{j=1}^J (l+1)^2 \sum_{i=1}^{I_j} n_i^j + (J-\beta-1)(l+1)^4 + n_p(l+1)^2, \quad (\text{D.6})$$

when the given cube's parent has no local expansion. If the given cube's parent has a local expansion, then (D.3) and (D.4) imply that the direct evaluation is the same or equal cost when

$$\sum_{j=1}^J n_p^j \sum_{i=1}^{I_j} n_i^j \leq \sum_{j=1}^J (l+1)^2 \sum_{i=1}^{I_j} n_i^j + (J-1)(l+1)^4. \quad (\text{D.7})$$

The inequalities (D.6) and (D.7) are alternative local-exactness checks to (D.1). In general (D.6) and (D.7) are easier to satisfy, so using them over (D.1) reveals more exact cubes. However, the added efficiency does not justify the extra cost of the (D.6) and (D.7) checks. The



minimum effective interaction list length,

$$I_{eff} \triangleq \left( \min_j \sum_{i=1}^{I_j} n_i^j \right) / (l+1)^2,$$

is the length of the lowest-cost interaction list if it prescribes only M2L operations. Thus if (D.6) holds then

$$\begin{aligned} I_{eff}(l+1)^2 \sum_{j=1}^J n_p^j &\leq J I_{eff}(l+1)^4 + J(l+1)^4 + [n_p - (\beta+1)(l+1)^2](l+1)^2; \\ I_{eff} \sum_{j=1}^J n_p^j &\leq J I_{eff}(l+1)^2 + J(l+1)^2 + n_p - (\beta+1)(l+1)^2. \end{aligned} \quad (D.8)$$

If level  $D$  is the lowest level and the given cube is on level  $d$ , then

$$\sum_{j=1}^J n_p^j = (D-d+1)n_p,$$

since the all non-empty cubes on each level must contain all  $n_p$  evaluation points in the given cube. Substitution into (D.8) and rearranging gives

$$[(D-d+1)I_{eff} - 1]n_p \leq (I_{eff} + 1)J(l+1)^2 - (\beta+1)(l+1)^2. \quad (D.9)$$

Similarly, if (D.7) holds then

$$(D-d+1)I_{eff}n_p \leq (I_{eff} + 1)J(l+1)^2 - (l+1)^2. \quad (D.10)$$

However, if all the  $J$  descendants of the given cube (this includes the cube itself) have at least  $(l+1)^2$  evaluation points<sup>1</sup>, then

$$(D-d+1)n_p \geq J(l+1)^2, \quad (D.11)$$

since all the cubes on a given level contain exactly the  $n_p$  evaluation points inside the given cube. For moderate values of  $I_{eff} \gg 1$ , both (D.9) and (D.10) contradict (D.11). Since  $I_{eff} \gg 1$  is a reasonable approximation in practice, the adaptive fast multipole algorithm uses the simpler local-exactness condition (D.1) rather than (D.6) and (D.7).

<sup>1</sup>This assumption does not affect the generality of the argument since it makes the direct-evaluation approach more expensive compared to the local-expansion method.

hol

de

ll h  
ode  
ona  
xac

---

---

# Bibliography

- [1] A. Ahmadouche and J. Chilo. Optimum computation of capacitance coefficients of multilevel interconnecting lines for advanced package. *IEEE Transactions on Components, Hybrids and Manufacturing Technology*, 12(1):124–129, March 1989.
- [2] M. S. A. Albouelwafa and E. J. M. Kendall. Analysis and design of helical capacitance sensors for volume fraction determination. *Review of Scientific Instruments*, 50(7):872–878, July 1979.
- [3] C. R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM Journal on Scientific and Statistical Computing*, 13(4):923–947, July 1992.
- [4] K. E. Atkinson. A survey of boundary integral equation methods for the numerical solution of laplace's equation in three dimensions. In M. A. Goldberg, editor, *Numerical Solution of Integral Equations*, pages 1–34. Plenum Press, New York, 1990.
- [5] A. K. Bahrani. Comment on "Solution of laplace's equation in a region with dielectric interfaces of arbitrary shape". *Proceedings of the IEEE*, 58(5):794–795, May 1970. (Letter).
- [6] P. Balaban. Calculation of the capacitance coefficients of planar conductors on a dielectric surface. *IEEE Transactions on Circuit Theory*, CT-20(6):725–731, November 1973.
- [7] K. J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [8] P. Benedek. Capacitances of a planar multiconductor configuration on a dielectric substrate by a mixed order finite-element method. *IEEE Transactions on Circuits and Systems*, CAS-23(5):279–284, May 1976.
- [9] P. Benedek and P. Silvester. Capacitance of parallel rectangular plates separated by a dielectric sheet. *IEEE Transactions on Microwave Theory and Techniques*, MTT-20(8):504–510, August 1972.

- [10] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics*, 44:141–183, 1991.
- [11] H. Blum. Numerical treatment of corner and crack singularities. In E. Stein and W. L. Wendland, editors, *Finite Element and Boundary Element Techniques from Mathematical and Engineering Point of View*, pages 171–212. Springer-Verlag, Wien, 1988.
- [12] R. R. Boix and M. Horno. Capacitance computation of elliptic microstrip disks in biaxial anisotropic multilayered substrates. *IEEE Transactions on Microwave Theory and Techniques*, MTT-38(1):30–37, January 1990.
- [13] C. W. Bostian and P. H. Wiley. Concerning the moment solution for the charge distribution on a square conducting plate. *Proceedings of the IEEE*, 59(11):1639, November 1971. (Letter).
- [14] C. A. Brebbia and J. Dominguez. *Boundary Elements: An Introductory Course*. Computational Mechanics Publications, Southampton, 1989. Co-published with McGraw-Hill, New York.
- [15] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel. *Boundary Element Techniques*. Springer-Verlag, Berlin, 1984.
- [16] C. M. Butler. Capacitance of a finite-length conducting cylindrical tube. *Journal of Applied Physics*, 51(11):5607–5609, November 1980.
- [17] B. L. Buzbee, F. W. Dorr, J. A. George, and G. H. Golub. The direct solution of the discrete poisson equation on irregular regions. *SIAM Journal on Numerical Analysis*, 8(4):722–736, December 1971.
- [18] J. B. Campbell and D. Makow. Calculation of the capacitance of a ring capacitor of rectangular cross section with no insulating gaps. *Journal of Computational Physics*, 12(1):137–142, May 1973. (Note).
- [19] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM Journal on Scientific and Statistical Computing*, 9(4):669–686, July 1988.
- [20] I. A. Cermak, P. Silvester, and S. K. Wong. Capacitive determination for infinite interdigital structures. *IEEE Transactions on Microwave Theory and Techniques*, 18(2):116, February 1970. (Computer program description).
- [21] J.-H. Chern, J. Huang, L. Arledge, P.-C. Li, and P. Yang. Multilevel metal capacitance models for CAD design synthesis systems. *IEEE Electron Device Letters*, 13(1):32–34, January 1992.

- [22] Y. L. Chow, Y. F. Lan, and D. G. Fang. Capacitance and its upper and lower bounds by the method of optimized simulated images. *Journal of Applied Physics*, 53(11):7144-7148, November 1982.
- [23] Y. L. Chow and M. M. Yovanovich. The shape factor of the capacitance of a conductor. *Journal of Applied Physics*, 53(12):8470-8475, December 1982.
- [24] P. Chowdhuri. Calculation of series capacitance for transient analysis of windings. *IEEE Transactions on Power Delivery*, 2(1):133-139, 1987.
- [25] K. Chun and K. D. Wise. A high-performance silicon tactile imager based on a capacitive cell. *IEEE Transactions on Electron Devices*, 32(7):1196-1201, July 1985.
- [26] J. C. Clements, C. R. Paul, and A. T. Adams. Computation of the capacitance matrix for systems of dielectric-coated cylindrical conductors. *IEEE Transactions on Electromagnetic Compatibility*, EMC-17(4):238-248, November 1975.
- [27] L. Collatz. *The Numerical Treatment of Differential Equations*. Springer-Verlag, New York, third edition, 1966.
- [28] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Company, New York, 1990.
- [29] P. E. Cottrell and E. M. Buturla. VLSI wiring capacitance. *IBM Journal of Research and Development*, 29(3):277-287, May 1985.
- [30] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience Publishers, New York, 1953.
- [31] S. H. Crandall. *Engineering Analysis*. McGraw-Hill, New York, 1956.
- [32] R. Dautray and J. L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology*, volume 2. Springer-Verlag, Berlin, 1988.
- [33] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, Orlando, Florida, 1984.
- [34] L. Debnath and P. Mikusiński. *Introduction to Hilbert Spaces with Applications*. Academic Press, Boston, 1990.
- [35] W. Delbale and D. De Zutter. Space-domain green's function approach to the capacitance calculation of multiconductor lines in multilayered dielectrics with improved surface charge modeling. *IEEE Transactions on Microwave Theory and Techniques*, 37(10):1562-1568, October 1989.

- [36] L. M. Delves and J. L. Mohamed. *Computational Methods for Integral Equations*. Cambridge University Press, Cambridge, 1985.
- [37] N. Deo. *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, Englewood Cliffs, New Jersey, 1974.
- [38] P. Dewilde and Z. Q. Ning. *Models for Large Integrated Circuits*. Kluwer Academic Publishers, Boston, 1990.
- [39] H. Engan. Surface acoustic wave multielectrode transducers. *IEEE Transactions on Sonics and Ultrasonics*, 22(6):395-401, November 1975.
- [40] V. A. Erma. Perturbation approach to the electrostatic problem for irregularly shaped conductors. *Journal of Applied Physics*, 4(12):1517-1526, December 1963.
- [41] A. Farrar and A. T. Adams. Matrix methods for microstrip three-dimensional problems. *IEEE Transactions on Microwave Theory and Techniques*, MTT-20(8):497-504, August 1972.
- [42] R. F. Fikhamanas and P. S. Fridberg. Howe method of calculating the capacitance of an object and its relation to variational principles. *Soviet Physics—Technical Physics*, 15(6):1027, December 1970.
- [43] B. A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic Press, New York, 1972.
- [44] S. Fukuda, N. Shigyo, K. Kato, and S. Nakamura. A ULSI 2-D capacitance simulator for complex structures based on actual processes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 9(1):39-47, January 1990.
- [45] J. A. Fuller and D. C. Chang. On the numerical calculation of capacitance in the presence of edge boundaries. *Proceedings of the IEEE*, 58(3):490-491, March 1970. (Letter).
- [46] P. R. Garabedian. *Partial Differential Equations*. John Wiley & Sons, New York, 1964.
- [47] I. Gladwell and R. Wait. *A Survey of Numerical Methods for Partial Differential Equations*. Clarendon Press, Oxford, 1979.
- [48] A. K. Goel. Electrode parasitic capacitances in self-aligned and deep-recessed GaAs MESFETs. *Solid-State Electronics*, 31(10):1471-1476, October 1988.
- [49] M. A. Golberg. A survey of numerical methods for integral equations. In M. A. Golberg, editor, *Solution Methods for Integral Equations*, pages 1-58. Plenum, New York, 1979.
- [50] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, second edition, 1989.

- [51] A. Greenbaum, L. Greengard, and G. B. McFadden. Laplace's equation and the dirichlet-neumann map in multiply connected domains. Mathematics and Computers DOE/ER/25053-5/UC-32, Courant Institute of Mathematical Sciences, New York University, New York, New York, March 1991.
- [52] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. M.I.T. Press, Cambridge, Massachusetts, 1988.
- [53] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, December 1987.
- [54] L. Greengard and V. Rokhlin. On the efficient implementation of the fast multipole algorithm. Technical Report YALEU/DCS/RR-602, Yale University, New Haven, Connecticut, February 1988.
- [55] R. Guerrieri and A. Sangiovanni-Vincentelli. Three-dimensional capacitance evaluation on a connection machine. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7(11):1125–1133, November 1988.
- [56] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54(4):463–491, 1989.
- [57] R. F. Harrington. *Field Computation by Moment Methods*. MacMillan, New York, 1968.
- [58] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, December 1952.
- [59] E. W. Hobson. *The Theory of Spherical and Ellipsoidal Harmonics*. Chelsea, New York, 1955.
- [60] R. W. Hockney. The potential calculation and some applications. In B. Alder, S. Fernbach, and M. Rotenberg, editors, *Methods in Computational Physics, Advances in Research and Applications, Plasma Physics*, pages 135–211. Academic Press, New York, 1970.
- [61] J. D. Jackson. *Classical Electrodynamics*. John Wiley & Sons, New York, second edition, 1975.
- [62] S. Jaffard. Wavelet methods for fast resolution of elliptic problems. *SIAM Journal of Numerical Analysis*, 29(4):965–986, August 1992.
- [63] M. A. Jaswon and G. T. Symm. *Integral Equation Methods in Potential Theory and Elastostatics*. Academic Press, London, 1977.
- [64] B. Johnson, S. Kim, S. D. Senturia, and J. White. MEMCAD capacitance calculations for mechanically deformed square diaphragm and beam microstructures. In *Proceedings of Transducers '91*, June 1991.

- [65] M. Kamon, M. J. Tsuk, and J. White. FastHenry: A multipole-accelerated 3-D inductance extraction program. To appear in *Proceedings of the 30th Design Automation Conference*, June 1993.
- [66] O. D. Kellogg. *Foundations of Potential Theory*. Dover Publications, New York, 1953. Reissue of the 1929 edition.
- [67] T. Kitazawa. Metalization thickness effect of striplines with anisotropic media—quasi-static and hybrid mode analysis. *IEEE Transactions on Microwave Theory and Techniques*, 37(4):769–775, April 1989.
- [68] Y. Konishi, M. Kumanoya, H. Yamasaki, K. Dosaka, and T. Yoshihara. Analysis of coupling noise between adjacent bit lines in megabit DRAM's. *IEEE Journal of Solid-State Circuits*, 24(1):35–42, February 1989.
- [69] R. Kress. *Linear Integral Equations*. Springer-Verlag, Berlin, 1989.
- [70] S. Kumar and D. Cho. A perturbation method for calculating the capacitance of electrostatic motors. In *Proceedings of the IEEE Micro Electro Mechanical Systems Conference*, pages 27–33, February 1990.
- [71] F. S. Lai. Coupling capacitances in VLSI circuits calculated by multi-dimensional discrete fourier series. *Solid-State Electronics*, 32(2):141–148, February 1989.
- [72] J. L. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problems and Applications*, volume 1. Springer-Verlag, New York, 1972.
- [73] S. C. Loh. Calculation of the electric field and the capacitance of a charged spherical bowl by means of toroidal co-ordinates. *Proceedings of the IEE*, 117(3):641–645, March 1970.
- [74] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachusetts, 1973.
- [75] W. Magnus and F. Oberhettinger. *Special Functions of Mathematical Physics*. Chelsea, New York, 1949.
- [76] J. E. Marsden and A. J. Tromba. *Vector Calculus*. W. H. Freeman and Company, San Francisco, 1976.
- [77] P. S. Maruvada and N. Hytenca. Capacitance calculation for some basic high-voltage electrode configurations. *IEEE Transactions on Power Apparatus and Systems*, 94(5):1708–1713, 1975.
- [78] W. E. Matzke, B. Heinemann, and G. Telschow. On the calculation of the capacitance coefficients for VLSI multilevel metalization lines by using domain methods. *IEEE Transactions on Electron Devices*, 36(2):447–449, February 1989. (Brief).



- [79] S. P. McCormick. EXCL: A circuit extractor for IC designs. In *Proceedings of the 22-nd ACM/IEEE Design Automation Conference*, pages 624-628, June 1984.
- [80] J. L. McCreary and P. R. Gray. All-MOS charge redistribution analog-to-digital conversion techniques—Part I. *IEEE Transactions on Circuits and Systems*, 10(12):371-379, December 1975.
- [81] M. Mehregany, P. Nagarkar, S. D. Senturia, and J. H. Lang. Operation of microfabricated harmonic and ordinary side-drive motors. In *Proceedings of the IEEE Micro Electro Mechanical Systems Conference*, pages 1-8, February 1990.
- [82] S. G. Mikhlin. *Integral Equations and Their Applications to Certain Problems in Mechanics, Mathematical Physics and Technology*. Pergamon Press, New York, 1957.
- [83] K. Nabors, S. Kim, and J. White. Fast capacitance extraction of general three-dimensional structures. *IEEE Transactions on Microwave Theory and Techniques*, 40(7):1496-1506, November 1992.
- [84] K. Nabors, S. Kim, J. White, and S. Senturia. Fast capacitance extraction of general three-dimensional structures. In *Proceedings of the 1991 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pages 479-484, October 1991.
- [85] K. Nabors, T. Kormeyer, and J. White. Multipole-accelerated preconditioned iterative methods for three-dimensional potential integral equations of the first kind. To appear in *SIAM Journal on Scientific and Statistical Computing*.
- [86] K. Nabors, T. Kormeyer, and J. White. Multipole-accelerated preconditioned iterative methods for three-dimensional potential integral equations of the first kind. In *Proceedings of the Copper Mountain Conference on Iterative Methods*, April 1992.
- [87] K. Nabors and J. White. A fast multipole algorithm for capacitance extraction of complex 3-D geometries. In *Proceedings of the IEEE 1989 Custom Integrated Circuits Conference*, pages 27.7.1-4, May 1989.
- [88] K. Nabors and J. White. Fastcap: A multipole accelerated 3-D capacitance extraction program. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(11):1447-1459, November 1991.
- [89] K. Nabors and J. White. An improved approach to including conformal dielectrics in multipole-accelerated three-dimensional capacitance extraction. In *Proceedings of the Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits: NU-PAD IV*, pages 167-172, May-June 1992.

- [90] K. Nabors and J. White. Multipole-accelerated 3-D capacitance extraction algorithms for structures with conformal dielectrics. In *Proceedings of the 29th ACM/IEEE Design Automation Conference*, pages 710–715, June 1992.
- [91] K. Nabors and J. White. Multipole-accelerated capacitance extraction algorithms for 3-D structures with multiple dielectrics. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 39(11):946–945, November 1992.
- [92] Y. Nakagome, M. Aoki, S. Ikenaga, M. Horiguchi, S. Kimura, Y. Kawamoto, and K. Itoh. The impact of data-line interference noise on DRAM scaling. *IEEE Journal of Solid-State Circuits*, 23(5):1120–1127, October 1988.
- [93] J. C. Nedelec. Curved finite element methods for the solution of singular integral equations on surfaces in  $R^3$ . *Computer Methods in Applied Mechanics and Engineering*, 8(1):61–80, April-May 1976.
- [94] J. N. Newman. Distributions of sources and normal dipoles over a quadrilateral panel. *Journal of Engineering Mathematics*, 20(2):113–126, 1986.
- [95] Z. Q. Ning and P. M. Dewilde. Spider: Capacitance modelling for VLSI interconnections. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7(12):1221–1228, December 1988.
- [96] Z. Q. Ning, P. M. Dewilde, and F. L. Neerhoff. Capacitance coefficients for VLSI multilevel metallization lines. *IEEE Transactions on Electron Devices*, ED-34(3):644–649, March 1987.
- [97] D. P. O’Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra and Its Applications*, 29:293–322, February 1980.
- [98] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, March 1982.
- [99] B. S. Panwar, A. B. Bhattacharya, and E. Dieulesaint. For computation of static capacitance of interdigital transducers in multilayered media. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, UFFC-33(4):416–420, July 1986.
- [100] P. Patel. Calculation of capacitance coefficients for a system of irregular finite conductors on a dielectric sheet. *IEEE Transactions on Microwave Theory and Techniques*, MTT-19(11):862–869, November 1971.
- [101] M. Politi, G. Macchiarella, and G. B. Stracca. Fast computation of static capacitance for periodic SAW transducers. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 35(6):708–710, November 1988.

- [102] H. Prashad. Theoretical evaluation of impedance, capacitance and charge accumulation on roller-bearings. *Wear*, 125(3):223-239, August 1, 1988.
- [103] A. Premoli. A new fast and accurate algorithm for the computation of microstrip capacitances. *IEEE Transactions on Microwave Theory and Techniques*, MTT-23(8):642-648, August 1975.
- [104] J. P. Quine, H. F. Webster, H. H. Glascock, II, and R. O. Carlson. Characterization of via connections in silicon circuit boards. *IEEE Transactions on Microwave Theory and Techniques*, 36(1):21-26, January 1988.
- [105] S. M. Rao, T. K. Sarkar, and R. F. Harrington. The electrostatic field of conducting bodies in multiple dielectric media. *IEEE Transactions on Microwave Theory and Techniques*, MTT-32(11):1441-1448, November 1984.
- [106] A. Ringhandt and H. G. Wagemann. An exact calculation of the two-dimensional capacitance of a wire and a new approximation formula. *IEEE Transactions on Electron Devices*, 40(5):1028-1032, May 1993.
- [107] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):187-207, September 15, 1985.
- [108] A. E. Ruehli. Survey of computer-aided electrical analysis of integrated circuit interconnections. *IBM Journal of Research and Development*, 23(6):626-639, November 1979.
- [109] 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.
- [110] A. E. Ruehli, P. A. Brennan, and H. W. Young. Recent progress in capacitance computation methods. In *Proceedings of the 1977 IEEE International Symposium on Circuits and Systems*, pages 135-138, Phoenix, April 1977.
- [111] Y. Saad and A. Sameh. A parallel block stiefel method for solving positive definite systems. In M. H. Schultz, editor, *Elliptic Problem Solvers*, pages 405-412. Academic Press, New York, 1981.
- [112] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856-869, July 1986.
- [113] J. J. Sakurai. *Modern Quantum Mechanics*. Addison-Wesley, Reading, Massachusetts, 1985.

- [114] T. Sakurai. Closed-form expressions for interconnection delay, coupling, and crosstalk in vlsi's. *IEEE Transactions on Electron Devices*, 40(1):118-124, January 1993.
- [115] M. R. Scheinfein and O. A. Palusinski. Methods of calculation of electrical parameters for electronic packaging applications. *Transactions of the Society for Computer Simulation*, 4(3):187-254, July 1987.
- [116] P. J. Schubert and J. H. Nevin. A polyimide-based capacitive humidity sensor. *IEEE Transactions on Electron Devices*, 32(7):1220-1223, July 1985.
- [117] A. Seidl, H. Klose, M. Svoboda, J. Oberndorfer, and W. Rösner. CAPCAL—a 3-d capacitance solver for support of CAD systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7(5):549-556, May 1988.
- [118] N. Shigyo, S. Fukuda, and K. Kato. The influence of boundary locations on wiring capacitance simulation. *IEEE Transactions on Electron Devices*, 36(6):1171-1174, June 1989.
- [119] P. Silvester and P. Benedek. Equivalent capacitances of microstrip gaps and steps. *IEEE Transactions on Microwave Theory and Techniques*, 20(11):729-733, November 1972.
- [120] P. Silvester and P. Benedek. Equivalent capacitances of microstrip open circuits. *IEEE Transactions on Microwave Theory and Techniques*, 20(8):511-516, August 1972.
- [121] P. Silvester and P. Benedek. Microstrip discontinuity capacitances for right-angle bends, t junctions, and crossings. *IEEE Transactions on Microwave Theory and Techniques*, 21(5):341-346, May 1973.
- [122] P. Silvester and M. S. Hsieh. Finite-element solution of 2-dimensional exterior-field problems. *Proceedings of the IEE*, 118(12):1743-1747, December 1971.
- [123] P. Silvester and M. S. Hsieh. Projective solution of integral equations arising in electric and magnetic field problems. *Journal of Computational Physics*, 8(1):73-82, August 1971.
- [124] R. Singh, S. P. Singh, and A. B. Bhattacharyya. A fast and area-efficiency (*sic.*) BWC array D/A and A/D conversion scheme. *IEEE Transactions on Circuits and Systems*, 36(6):912-916, June 1989.
- [125] P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 10(1):36-52, January 1989.
- [126] I. Stakgold. *Boundary Value Problems of Mathematical Physics*, volume 2. Macmillan, New York, 1968.

- [127] G. Strang. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich, San Diego, third edition, 1988.
- [128] A. H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall, Englewood Cliffs, New Jersey, 1971.
- [129] H. Tanigawa, T. Ishihara, M. Hirata, and K. Suzuki. MOS integrated silicon pressure sensor. *IEEE Transactions on Electron Devices*, 32(7):1191-1195, July 1985.
- [130] C. D. Taylor, G. N. Elkhouri, and T. E. Wade. On the parasitic capacitances of multilevel skewed metallization lines. *IEEE Transactions on Electron Devices*, ED-33(1):41-46, January 1986.
- [131] L. N. Trefethen. Approximation theory and numerical linear algebra. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation II*, pages 336-360. Chapman and Hall, London, 1988.
- [132] L. A. Tseitlin. Calculation of the electric field and capacitance of elongated and flattened bodies. *Soviet Physics - Technical Physics*, 14(7):872-876, January 1970.
- [133] M. Uehara. Green's functions and coefficients of capacitance. *American Journal of Physics*, 54(2):184-185, February 1986.
- [134] N. P. van der Meijs and A. J. van Genderen. An efficient finite element method for submicron IC capacitance extraction. In *Proceedings of the 26-th ACM/IEEE Design Automation Conference*, pages 678-681, June 1989.
- [135] S. A. Vavasis. Preconditioning for boundary integral equations. *SIAM Journal of Matrix Analysis and Applications*, 13(3):905-925, July 1992.
- [136] A. Venema, J. J. M. Dekkers, and R. F. Humphrye. Static capacitance calculation for a surface acoustic wave interdigital transducer in multilayered media. *IEEE Transactions on Microwave Theory and Techniques*, 26(4):294-297, April 1978.
- [137] T. Wang, R. F. Harrington, and J. R. Mautz. Quasi-static analysis of a microstrip via through a hole in a ground plane. *IEEE Transactions on Microwave Theory and Techniques*, 36(6):1008-1013, June 1988.
- [138] W. T. Weeks. Calculation of coefficients of capacitance of multiconductor transmission lines in the presence of a dielectric interface. *IEEE Transactions on Microwave Theory and Techniques*, MTT-18(1):35-43, January 1970.
- [139] W. L. Wendland. Boundary element methods and their asymptotic convergence. In P. Filippi, editor, *Theoretical Acoustics and Numerical Techniques*, pages 135-216. Springer-Verlag, Wein, 1983.

- [140] W. L. Wendland. On asymptotic error estimates for combined BEM and FEM. In E. Stein and W. L. Wendland, editors, *Finite Element and Boundary Element Techniques from Mathematical and Engineering Point of View*, pages 271–333. Springer-Verlag, Wien, 1988.
- [141] W. Whitehead. The role of computer models in connector standardization. In *24th Annual Connector and Interconnection Technology Symposium*, pages 103–110, San Diego, CA, 1991.
- [142] R. B. Wu. Interperiod capacitance calculations for three-dimensional, multiconductor systems. *IEEE Transactions on Microwave Theory and Techniques*, 36(11):1515–1520, November 1988.
- [143] R. B. Wu and L. L. Wu. Exploiting structure periodicity and symmetry in capacitance calculations for three-dimensional multiconductor systems. *IEEE Transactions on Microwave Theory and Techniques*, 36(9):1311–1318, September 1988.
- [144] M. Yoshida, T. Takeshima, and M. Takada. Scaled bit line capacitance analysis using a three-dimensional simulator. In *1985 Symposium on VLSI Technology, Digest of Technical Papers*, pages 66–67, May 1985.
- [145] A. H. Zemanian, R. P. Tewarson, C. P. Ju, and J. F. Jen. Three-dimensional capacitance computations for VLSI/ULSI interconnections. *IEEE Transactions on Computer-Aided Design*, 8(12):1319–1326, December 1989.