A Multiscale Method for Fast Capacitance Extraction

Johannes Tausch
Dept. of Mathematics
Southern Methodist University
Dallas, TX 75275-0156
tausch@mail.smu.edu

Jacob White
Department of EECS
MIT
Cambridge, MA 02139
white@mit.edu

Abstract

The many levels of metal used in aggressive deep submicron process technologies has made fast and accurate capacitance extraction of complicated 3-D geometries of conductors essential, and many novel approaches have been recently developed. In this paper we present an accelerated boundary-element method, like the well-known FASTCAP program, but instead of using an adaptive fast multipole algorithm we use a numerically generated multiscale basis for constructing a sparse representation of the dense boundary-element matrix. Results are presented to demonstrate that the multiscale method can be applied to complicated geometries, generates a sparser boundary-element matrix than the adaptive fast multipole method, and provides an inexpensive but effective preconditioner. Examples are used to show that the better sparsification and the effective preconditioner yield a method that can be 25 times faster than FASTCAP while still maintain accuracy in the smallest coupling capacitances.

1 Introduction

The development of deep submicron processes with extremely fast devices and five or more levels of metal have made designers of high-performance digital and analog integrated progressively more concerned with extracting accurate self and coupling capacitances from complicated 3-D structures. Recently developed efficient algorithms for performing 3-D capacitance extraction have focussed on three techniques, the floating random walk method [10], improvements to the finite-difference and finite-element methods [3, 2] and the so-called fast methods based on acceleration of the method-of-moments or boundary-element approach [8, 9, 7].

In this paper we present a new multiscale, or wavelet-like, approach to accelerating the boundary-element method, and demonstrate the method on several examples. We show that this method has two important features: it can accurately represent the N^2 entries of the dense boundary-element matrix in provably order N elements with a low constant factor, and the method generates an inexpensive and extremely effective preconditioner. As has become traditional in this subject, we compare our results to the publically available Fastcap program, which uses a multipole-accelerated boundary-element method, and show that our method can be more than twenty-five times faster *without* significantly compromising accuracy even in small coupling capacitances.

In Section 2 we present a brief background on boundary-element methods and acceleration techniques. Then in the following sections we describe the construction of the multiscale basis and our matrix sparsification schemes. Finally, we make extensive comparisons between our multiscale method and the FASTCAP program for several examples. We show that without truncation, the multiscale method and the adaptive fast multipole algorithm generate sparse representations of the dense boundary-element matrix in *exactly* the same number of elements. We then show that truncation, which can only be used with the multiscale method, provides an additional factor of five over FASTCAP without compromising accuracy even in smallest coupling capacitances, and that truncation can be used to achieve a reduction of nearly a factor of 60, but then the self and larger coupling capacitances are accurate only to ten percent. We also show that the multiscale preconditioner reduces the cost of solving the boundary-element equations by as much as a factor of five, and when combined with the better sparsification gives a speed improvement over FASTCAP of anywhere from 25 to 300.

2 Background

The most commonly used integral formulation for computing the charge density σ of conductors for the given surface potential ψ is the first-kind integral equation

$$\mathcal{V}\sigma(x) = \psi(x)\,, \quad x \in \text{surfaces}, \tag{1}$$

where $\mathcal{V}\sigma(x)$ denotes the potential due to the charge distribution evaluated at an arbitrary fieldpoint x

$$\mathcal{V}\sigma(x) = \int_{\text{surfaces}} \frac{1}{4\pi\varepsilon_0\|x-x'\|}\sigma(x')\,dS_{x'}\,, \tag{2}$$

where $dS_{x'}$ is the incremental conductor surface area and $\|x\|$ denotes the usual Euclidean length.

One standard approach to numerically solving (1) is to use a piece-wise constant Galerkin scheme, where σ is represented by a set of uniformly-charged panels. The result is a linear system,

$$Aq = \overline{p} \tag{3}$$

where $P \in \mathbf{R}^{n\times n}$, q is the vector of panel charges, $\overline{p} \in \mathbf{R}^n$ is the vector of known panel potential averages,

$$A_{ij} = \langle \chi_i, \mathcal{V}\chi_j \rangle\,, \tag{4}$$

DAC 99, New Orleans, Louisiana: (c) 1999 ACM 1-58113-109-7/99/06..$5.00

χ_i is a function which is unity on panel i and zero elsewhere and $\langle , \rangle$ is the L_2-inner product.

Since every charge in the problem contributes to the potential everywhere, the system of equations for the unknown charge densites is dense, and a straight-forward implementation using Gaussian elimination requires order N^3 operations. It is much more efficient to solve the system iteratively, e.g., by conjugate gradients (cg), but the memory and cpu time required still scales superlinearly, roughly as N^2.

In the last decade there have been an number of approaches developed that can be used to reduce the computation time and memory required to solve (4) to nearly order N. These methods approximately and implicitly represent the dense matrix P in a sparse way, using much less than N^2 memory. The implicit representations can be used to compute the matrix-vector products required in cg in much less than N^2 time.

Almost all the sparsification techniques in use rely on the observation that nearby panel interactions must be represented accuractly and in detail, but distant panel interactions can be clustered together to achieve efficiency. The specific techniques that have been used include the adaptive fast multipole method [11, 13], the precorrected-FFT method [4], a hierarchical singular-value decomposition method [5] and a hierarchical panel clustering method [1].

Wavelets have also been applied to this problem [6] but the approaches have been restricted to problems with only one, or perhaps a few, surfaces and therefore have not been practical for capacitance extraction.

Similar to wavelets, our approach generates basisfunctions on multiple levels. On the finest level, weighted combinations of panels are used to generate two types of basis functions: The first type (later on denoted by ψ) have rapidly decaying potentials, and the second type (denoted by ϕ) is orthogonal to the ψ's. On the coarser levels, ϕ-functions of the next finest level are combined to form new basisfunctions with rapidly decaying potentials and their orthogonal complement.

As we will demonstrate below, our multiscale approach has two main benefits: it generates a very sparse representation of P even for extremely complicated geometries and it also provides an effective preconditioner for the iterative solver.

3 Construction of the Multiscale Basis

For the construction of the multiscale basis we will apply two concepts known from the Fast Multipole Algorithm.

- The *truncated multipole expansion* centered at the point x_0 of the characteristic function χ_j can be used to approximate the potential due to χ_j. It is given by

$$\mathcal{V}\chi_j(x) \approx \sum_{n=0}^{p} \sum_{m=-n}^{n} \mu_n^m(\chi_j) \frac{Y_n^m(\phi,\theta)}{\rho^{n+1}} \tag{5}$$

  Here ρ, ϕ, θ denote the spherical coordinates of the vector $x - x_0$, Y_n^m are the surface spherical harmonics and $d = \mathrm{dist}(S_j, x_0)$. The multipole coefficients $\mu_m^n(\chi_j)$ are given by

$$\mu_n^m(\chi_j) = \int_S r^n Y_n^{-m}(\alpha,\beta)\, \chi_j(r,\alpha,\beta)\, dS_{(r,\alpha,\beta)}\,. \tag{6}$$

  For flat panels the multipole coefficients can be calculated in closed form. For more details we refer to [13].

- *Hierarchical decomposition of the problem domain.* Embed the surface S into a coarsest level cube. The cube is subdivided into eight cubes of equal size and this process is iterated until the cubes in the finest level L contain at most a predetermined number of panels. The cubes at the l-th refinement level are collectively denoted by $C^{(l)}, l = 0, \ldots, L$.

The calculation of the multiscale basis follows the cube hierarchy from the finest to the coarsest level. At first, for every nonempty finest level cube $\nu \in \mathcal{C}^{(L)}$ the canonical basis functions $\{\chi_{\nu,1}, \ldots, \chi_{\nu,n}\}$ in ν are transformed to form a new orthogonal basis. This is done in a way such that multipole coefficients of order $\leq p$ of some of the newly formed basis functions vanish. The actual calculation will be described below in more detail. The basis functions with vanishing multipole coefficients are denoted by $\psi_\nu = \{\psi_{\nu,1}, \ldots, \psi_{\nu,s_\nu}\}$, and the remaining orthogonal basis function are denoted by $\phi_\nu = \{\phi_{\nu,1}, \ldots, \phi_{\nu,r_\nu}\}$.

In the new basis the matrix of the discretized operator assumes the form

$$\widehat{A}^{(L)} = \begin{bmatrix} \langle \phi^{(L)}, \mathcal{V}\phi^{(L)} \rangle & \langle \phi^{(L)}, \mathcal{V}\psi^{(L)} \rangle \\ \langle \psi^{(L)}, \mathcal{V}\phi^{(L)} \rangle & \langle \psi^{(L)}, \mathcal{V}\psi^{(L)} \rangle \end{bmatrix}. \tag{7}$$

where $\phi^{(L)}, \psi^{(L)}$ denote all level-L ϕ- and ψ-functions, respectively.

The functions $\psi_{\nu i}$ have vanishing multipole expansions and because of (5) their potentials $\mathcal{V}\psi_{\nu i}$ are rapidly decreasing functions. Therefore the entries

$$\langle \phi_{\nu' i'}, \mathcal{V}\psi_{\nu i} \rangle\,,\ \langle \psi_{\nu' i'}, \mathcal{V}\phi_{\nu i} \rangle\,,\ \langle \psi_{\nu' i'}, \mathcal{V}\psi_{\nu i} \rangle \tag{8}$$

will be neglected if ν and ν' are not neighboring cubes. Hence three blocks of the matrix $\widehat{A}^{(L)}$ are sparse.

Once the finest level transform has been achieved, next level functions $\phi^{(L-1)}$ and $\psi^{(L-1)}$ are obtained by transforming the $\phi^{(L)}$'s in a way such that the newly formed $\psi^{(L-1)}$-functions have vanishing multipole coefficients. This process is iterated until a basis of the form $\{\psi^{(L)}, \psi^{(L-1)}, \ldots, \psi^{(2)}, \phi^{(2)}\}$ has been constructed. We describe the transformation below in more detail.

Finest Level

First consider the finest level

cube $\nu \in \mathcal{C}^{(L)}$ that contans n_ν panels. The new basis is formed by combining the characteristic functions of the panels

$$\phi_{\nu,i} = \sum_j q_{i,j}\, \chi_{\nu,j}, \quad i = 1, \ldots, r_\nu \tag{9}$$

$$\psi_{\nu,i} = \sum_j q_{i+r_\nu,j}\, \chi_{\nu,j}, \quad i = 1, \ldots, s_\nu\,, \tag{10}$$

where $r_\nu + s_\nu = n_\nu$. The ϕ- and the ψ-functions are linear combinations of the χ-functions and so are their moments. In matrix form this relation takes the form

$$[\mu(\phi), \mu(\psi)] = M_\nu Q_\nu^T\,, \tag{11}$$

where $M_\nu \in \mathbf{R}^{(p+1)^2 \times n_\nu}$ is the matrix whose i-th column contains the moments of $\chi_{\nu,i}$, $\mu(\phi) \in \mathbf{R}^{(p+1)^2 \times r_\nu}$ is the matrix that contains the moments of $\phi_{\nu,i}$, $\mu(\psi) \in \mathbf{R}^{(p+1)^2 \times s_\nu}$ is the matrix that contains the moments of $\psi_{\nu,i}$ and Q_ν contains the coefficients $q_{i,j}$.

The transformation Q_ν that makes the multipole moments of the ψ-functions in vanish comes from the singular value decomposition

$$M_\nu = U_\nu S_\nu Q_\nu^T \tag{12}$$

of M_ν. Here, $U_\nu = u_{i,j}$ is unitary and S_ν is a diagonal matrix with non-zero singular values $s_1, \ldots, s_{r_\nu}$.

If the coefficients in (9) and (10) are the entries of the matrix Q_ν in (12), then the ϕ-functions have vanishing multipole moments and are orthogonal to the ψ-functions. To determine the new basis in the higher levels we will need the multipole moments of the ϕ-functions which are given by

$$\mu_n^m(\phi_{\nu,i}) = s_i u_{(n,m),i}\,, \quad i = 1, \ldots, r_\nu\,. \tag{13}$$

Higher levels

Suppose multiscale basis functions $\phi^{(k)}, \psi^{(k)}, k = L, \ldots, l+1$ have been constructed. Consider now the non-empty cube $\nu \in \mathcal{C}^{(l)}$. The new basis functions ϕ_ν, ψ_ν are linear combinations of the ϕ-functions of cube ν's children $\alpha \in \mathcal{K}_\nu$

$$\phi_{\nu,i} = \sum_{j,\alpha} q_{i,(\alpha,j)} \phi_{\alpha,j}, \quad i = 1, \ldots, r_\nu \tag{14}$$

$$\psi_{\nu,i} = \sum_{j,\alpha} q_{i+r_\nu,(\alpha,j)} \phi_{\alpha,j}, \quad i = 1, \ldots, s_\nu\,, \tag{15}$$

This leads again to the singular value decomposition of M_ν as in (12). The matrix M_ν contains the multipole moments of the functions $\phi_\alpha, \alpha \in \mathcal{K}_\nu$ centered in x_ν. These moments have been computed in the previous levels, however, they are centered in the x_α's and must be translated to the new center x_ν. This is a linear transformation which also arises in the Fast Multipole algorithm . It has been described by Greengard [11].

Multiresolution Transform

The algorithm described above generates, beginning with the characteristic functions at the finest level, a sequence of orthogonal bases of the finite element space

$$\begin{aligned} \{\chi_1, \ldots, \chi_N\} &\to \left\{\phi^{(L)}, \psi^{(L)}\right\} \\ &\to \left\{\phi^{(L-1)}, \psi^{(L-1)}, \psi^{(L)}\right\} \\ &\cdots \\ &\to \left\{\phi^{(2)}, \psi^{(2)}, \ldots, \psi^{(L-1)}, \psi^{(L)}\right\}\,. \end{aligned}$$

A piecewise constant charge distribution can be expanded in the canonical basis as well as one of the multiscale bases at any level

$$\sigma_h = \sum_{i=1}^{N} \sigma_i \chi_i = \sum_{k=L}^{l} \sum_{\nu \in \mathcal{C}_k} \sum_{i=1}^{s_\nu} \widehat{\sigma}_{\nu,i} \psi_{\nu,i} + \sum_{\nu \in \mathcal{C}_k} \sum_{i=1}^{r_\nu} \widetilde{\sigma}_{\nu,i} \phi_{\nu,i}$$

where

$$\begin{aligned} \sigma_i &= \langle \sigma_h, \chi_i \rangle \\ \widehat{\sigma}_{\nu,i} &= \langle \sigma_h, \psi_{\nu,i} \rangle \\ \widetilde{\sigma}_{\nu,i} &= \langle \sigma_h, \phi_{\nu,i} \rangle\,. \end{aligned}$$

Defining the vectors $\widehat{\sigma}_\nu = [\widehat{\sigma}_{\nu,i}]_i, \widetilde{\sigma}_\nu = [\widetilde{\sigma}_{\nu,i}]_i, \widehat{\sigma}^{(l)} = [\widehat{\sigma}_\nu]_{\nu \in \mathcal{C}_l}$ and $\widetilde{\sigma}^{(l)} = [\widetilde{\sigma}_\nu]_{\nu \in \mathcal{C}_l}$ the transform of σ_h can be written as

$$\begin{array}{ccccccccc} \sigma & \to & \widetilde{\sigma}^{(L)} & \to & \widetilde{\sigma}^{(L-1)} & \to & \ldots & \to & \widetilde{\sigma}^{(2)} \\ & \searrow & & \searrow & & \searrow & & \searrow & \\ & & \widehat{\sigma}^{(L)} & & \widehat{\sigma}^{(L-1)} & & \ldots & & \widehat{\sigma}^{(2)} \end{array} \tag{16}$$

which can be viewed the multiresolution analysis of the function σ. Since there are only local transformations the calculation of $\widehat{\sigma}$ takes order N operations.

4 Calculation of the Non-Standard Form

Because of the multiresolution analysis the machinery of compressing integral operators expressed in the multiscale basis is quite similar to the compession with wavelet bases described by Beylkin et al. [6]. Similar to the wavelet case, there are two alternatives to carry out matrix-vector multiplications in the multiscale basis.

The first alternative is to transform the discretized linear system into the multiscale basis

$$\widehat{A}\widehat{x} = \widehat{b}, \tag{17}$$

where $\widehat{A}$ is the transformation of the stiffness matrix, or standard form and is given by

$$\widehat{A} = \begin{bmatrix} \langle \phi^{(2)}, \mathcal{V}\phi^{(2)} \rangle & \langle \phi^{(2)}, \mathcal{V}\psi \rangle \\ \langle \psi, \mathcal{V}\phi^{(2)} \rangle & \langle \psi, \mathcal{V}\psi \rangle \end{bmatrix}. \tag{18}$$

Here $\mathcal{V}$ denotes the integral operator in (2). The second alternative is to use the non-standard form

$$\begin{aligned} \widehat{A}_{\text{ns}} = \text{BlockDiag} \Bigg(& \begin{bmatrix} \langle \phi^{(2)}, \mathcal{V}\phi^{(2)} \rangle & \langle \phi^{(2)}, \mathcal{V}\psi^{(2)} \rangle \\ \langle \psi^{(2)}, \mathcal{V}\phi^{(2)} \rangle & \langle \psi^{(2)}, \mathcal{V}\psi^{(2)} \rangle \end{bmatrix}, \\ & \begin{bmatrix} 0 & \langle \phi^{(3)}, \mathcal{V}\psi^{(3)} \rangle \\ \langle \psi^{(3)}, \mathcal{V}\phi^{(3)} \rangle & \langle \psi^{(3)}, \mathcal{V}\psi^{(3)} \rangle \end{bmatrix}, \ldots \\ & \ldots, \begin{bmatrix} 0 & \langle \phi^{(L)}, \mathcal{V}\psi^{(L)} \rangle \\ \langle \psi^{(L)}, \mathcal{V}\phi^{(L)} \rangle & \langle \psi^{(L)}, \mathcal{V}\psi^{(L)} \rangle \end{bmatrix} \Bigg)\,. \end{aligned} \tag{19}$$

For a matrix-vector product in the non-standard form, all components of the charge vector in the multiscale basis

$$\sigma_{\text{ns}} = \left(\widetilde{\sigma}^{(l)}, \widehat{\sigma}^{(l)}\right)_{l=2,\ldots,L}$$

are calculated via the multiresolution analysis (16). In a second step, the vector $\widehat{u}_{\text{ns}} = \widehat{A}_{\text{ns}}\widehat{\sigma}_{\text{ns}}$ is formed and finally the vector in the original basis is obtained by the additive inverse transformation

$$\begin{array}{ccccccccc} \widetilde{u}_{\text{ns}}^{(2)} & \to & \widetilde{u}_{\text{ns}}^{(3)} & \to & \ldots & \to & \widetilde{u}_{\text{ns}}^{(L)} & \to & u \\ & \nearrow & & \nearrow & & \nearrow & & \nearrow & \\ \widehat{u}_{\text{ns}}^{(2)} & & \widehat{u}_{\text{ns}}^{(3)} & & \ldots & & \widehat{u}_{\text{ns}}^{(L)} & & \end{array} \tag{20}$$

Here the transforms of the higher levels must be *added* to the already computed ϕ-components of the potential vector.

Because of the rapid decay of the potentials $\mathcal{V}\psi$, the matrix coefficients $\langle \phi_\nu, \mathcal{V}\psi_{\nu'} \rangle$, $\langle \psi_\nu, \mathcal{V}\phi_{\nu'} \rangle$ and $\langle \psi_\nu, \mathcal{V}\psi_{\nu'} \rangle$ are dropped whenever ν and ν' are not neighboring cubes. Thus both, the standard as well as the non-standard form are approximated by sparse matrices.

Contrary to the standard form, the non-standard form does not contain interactions of ψ-functions at different levels. Thus, for a fixed polynomial degree, the non-standard form contains order N non-zero terms. For the standard form truncation strategies can be developed that also leave only order N terms. However, the non-standard is somewhat easier to implement, and has been used in our implementation. In the remainder of this section we describe how the coefficients in the sparsified non-standard form can be calculated efficiently.

Finest Level

In view of (9) and (10) the bottom level ϕ and ψ's are linear combinations of the canonical basis functions. Thus the interactions of ϕ and ψ's of neighboring cubes $\nu' \in \mathcal{N}_\nu$ are transformations of the χ's

$$\begin{bmatrix} \langle \phi_{\nu'}, \mathcal{V}\phi_\nu \rangle & \langle \phi_{\nu'}, \mathcal{V}\psi_\nu \rangle \\ \langle \psi_{\nu'}, \mathcal{V}\phi_\nu \rangle & \langle \psi_{\nu'}, \mathcal{V}\psi_\nu \rangle \end{bmatrix} = Q_\nu \langle \chi_\nu, \mathcal{V}\chi_{\nu'} \rangle Q_{\nu'}^T .$$

For the calculation of the blocks $\langle \chi_\nu, \mathcal{V}\chi_{\nu'} \rangle$ standard quadrature rules can be used. The blocks $\langle \phi_\nu, \mathcal{V}\phi_{\nu'} \rangle$ do not appear in the non-standard form, but they will be needed to set up the matrix coefficients of the next higher levels. Since the farfield is dropped in the non-standard form, only interactions between neighboring cubes must be calculated.

Coarser Levels

Suppose that the coefficients $\langle \phi_\nu, \mathcal{V}\phi_{\nu'} \rangle$, $\langle \phi_\nu, \mathcal{V}\psi_{\nu'} \rangle$, $\langle \psi_\nu, \mathcal{V}\phi_{\nu'} \rangle$ and $\langle \psi_\nu, \mathcal{V}\psi_{\nu'} \rangle$ have been calculated for all $\nu' \in \mathcal{N}_\nu, \nu \in \mathcal{C}^(k), k > l$. For a given cube ν in level l, the ϕ_ν and ψ_ν's are linear combinations of the $\phi_\alpha, \alpha \in \mathcal{K}_\nu$ as in (14) and (15). Thus the entries in the non-standard form are transformations of ϕ-functions of the children cubes

$$\begin{bmatrix} \langle \phi_\nu, \mathcal{V}\phi_{\nu'} \rangle & \langle \phi_\nu, \mathcal{V}\psi_{\nu'} \rangle \\ \langle \psi_\nu, \mathcal{V}\phi_{\nu'} \rangle & \langle \psi_\nu, \mathcal{V}\psi_{\nu'} \rangle \end{bmatrix} = Q_\nu \Big[\langle \phi_\alpha, \mathcal{V}\phi_{\alpha'} \rangle \Big]_{\alpha \in \mathcal{K}_\nu \, \alpha' \in \mathcal{K}_{\nu'}} Q_{\nu'}^T .$$

For $\alpha' \in \mathcal{N}_\alpha$ the entry $\langle \phi_\alpha, \mathcal{V}\phi_{\alpha'} \rangle$ of the matrix in the right hand side has been calculated in the previous level. However, there are also interactions between cubes α and α' which are not neighbors but whose parents are neighbors. These are interactions between well-separated cubes and hence their calculation may be approximated using the multipole expansions. For that, set $x = x_\alpha + h$, $y = x_{\alpha'} + k$ and $r_{\alpha,\alpha'} = x_\alpha - x_{\alpha'}$, then

$$\langle \phi_\alpha, \mathcal{V}\phi_{\alpha'} \rangle_{i,i'} = \int_{S_\alpha} \int_{S_{\alpha'}} \frac{\phi_{\alpha,i}(x)\phi_{\alpha',i'}(y)}{|r_{\alpha,\alpha'} + h - k|} \, dS_y \, dS_x .$$

The variables in the integrand can be separated using the translation theory of spherical harmonics. Thus the integral can be expressed in terms of the multipole moments of the ϕ-functions in cubes α, α'

$$\langle \phi_\alpha, \mathcal{V}\phi_{\alpha'} \rangle_{i,i'} \approx \mu(\phi_{\alpha i})^T L \mu(\phi_{\alpha' i'})$$

where L is the multipole-to-local translation matrix of cube α to cube α' of the Fast Multipole alogrithm, see [11].

5 Truncation of nearby interactions

The truncated non-standard form has non-zero terms only for interactions corresponding to neighboring cubes. These matrices have a large number of small entries which can be dropped to achieve a higher compression of the integral operator.

To understand why nearby interaction matrices have small entries, consider an arbitrary entry

$$\langle \psi_{\nu i}, \mathcal{V}\phi_{\nu' i'} \rangle$$

of two adjacent cubes ν, ν'. This entry is the inner product of the function $\psi_{\nu i}$ with the potential due to the function $\phi_{\nu' i'}$. This potential is a smooth harmonic function and furthermore, since the ψ-function has vanishing multipole coefficients, it is orthogonal to the solid spherical harmonics. Thus for any harmonic polynomial h_p of degree $\leq p$ we have $\langle \psi_{\nu i}, h_p \rangle = 0$ and

$$\langle \psi_{\nu i}, \mathcal{V}\phi_{\nu' i'} \rangle = \sup_{h_p} \langle \psi_{\nu i}, \mathcal{V}\phi_{\nu' i'} - h_p \rangle .$$

This quantity is small if the potential due to $\phi_{\nu' i'}$ can be well approximated by harmonic polynomials. In practice, it is difficult to predict *a priori* for which pairs of functions the interaction is small enough to be truncated from the matrix. Instead, every entry of nearby interaction matrices must be calculated and dropped if its modulus falls below a given threshold.

The threshold must be chosen so that the error due to truncating nearby interactions is not bigger than the error due to truncating distant interactions. In our implementation, we eliminate entries that satisfy

$$|a_{i,j}| \leq \varepsilon_T \frac{2^{-p}}{(p+1)^2 L} , \tag{21}$$

where p is the expansion order, L is the number of levels in the cube hierarchy and ε_T is a user-specified parameter. With this choice the additional error of the potential is $\mathcal{O}(2^{-p})$ and therefore decays exponentially with the expansion order.

6 Preconditioning

Multiscale bases bear a strong resemblence to the hierarchical bases schemes [12]. These schemes were originally designed to improve the convergence of iterative solvers for discretized PDEs, but they generalize to a large class of differential as well as integral operators. Their striking feature is that their representation with respect to the multiscale basis is diagonal except for a small block in the highest level.

The form of the standard form $\widehat{A}$ suggests a similar preconditioning strategy. The entries $\langle \Psi_{\nu' i'}, \mathcal{V}\Psi_{\nu i} \rangle$ are small not only for cubes at positive distances but also for intersecting cubes at different levels. Thus the diagonal blocks of $\widehat{A}$ have the largest impact on the matrix vector product and are used for preconditioning.

$$\widehat{P} = \text{BlockDiag} \left(\langle \Phi^{(2)}, \mathcal{V}\Phi^{(2)} \rangle, \langle \Psi_\nu, \mathcal{V}\Psi_\nu \rangle_{\nu \in \mathcal{C}_2 \cup \ldots \cup \mathcal{C}_L} \right) \tag{22}$$

It is more desirable to solve the discretized system in the original coordinates, where the more efficient non-standard form can be used for the matrix-vector product. Since the preconditioner is only given in the transformed basis, each preconditioning step involves an additional forward and an additional inverse multiscale transform.

7 Numerical Results

This section reports some numerical results obtained by using the multiscale basis described above. We have used Galerkin discretization with piecewise constant elements. The computations are based on the non-standard form (19), where all entries are truncated unless they correspond to nearest neighbor cubes.

We illustrate the convergence behavior of the compression scheme, the performance of the preconditioner, as well as the sparsity of the matrix on two example domains, the ellipsoid and the bus crossing structure of [7] with varying numbers conductors.

Accuracy

For the ellipsoid the the charge density is known analytically and compared with the the numerical solution for various expansion orders. The mean-square (L_2-)error of the charge density as a function on the surface, as well as the error of the capacitance are shown in Table 1. The L_2-error is halved if the meshwidth is halved and the expansion order increased by one. If only the capacitance is sought, high accuracies can be obtained with moderate expansion orders.

	L_2-error				
Panels	192	768	3072	12288	49152
$p=1$	0.4027	0.5041	0.7930	1.3416	2.0734
$p=2$	0.3591	0.1843	0.2400	0.4145	0.6151
$p=3$	0.3533	0.1513	0.0838	0.0841	0.1081
$p=4$	0.3530	0.1494	0.0721	0.0433	0.0352
$p=5$	0.3530	0.1500	0.0712	0.0362	0.0206
	Error Capacitance				
$p=1$	0.5916	0.1394	0.0049	0.0315	0.0419
$p=2$	0.5911	0.1531	0.0390	0.0095	0.0030
$p=3$	0.5799	0.1525	0.0387	0.0103	0.0041
$p=4$	0.5793	0.1508	0.0350	0.0058	0.0012
$p=5$	0.5793	0.1508	0.0348	0.0055	0.0016

Table 1: Discretization errors, ellipsoid. The exact value of the capacitance is 24.702.. No truncation of nearby terms.

For the bus-crossing structure, no analytical solution is known. To illustrate the convergence behavior of the truncation scheme, we show in Table 2 the self- and coupling for a fixed discretization and increasing expansion order. For the self-capacitance a low-order expansion will suffice. If the smallest coupling capacitances are required to a high accuracy, larger expansion orders are needed.

order	1	2	3	4	5
$C_{1,1}$	82.628	82.007	81.875	81.930	81.956
$C_{1,2}$	-29.11	-28.49	-28.69	-28.67	-28.68
$C_{1,3}$	-2.183	-2.340	-2.259	-2.284	-2.276
$C_{1,4}$	-0.533	-1.237	-1.059	-1.024	-1.027
$C_{1,5}$	-1.095	-0.361	-0.555	-0.619	-0.621
$C_{1,6}$	-0.878	-0.498	-0.432	-0.432	-0.433
$C_{1,7}$	-0.151	-0.319	-0.339	-0.341	-0.343
$C_{1,8}$	-0.510	-0.477	-0.456	-0.455	-0.455
$C_{1,9}$	-5.646	-5.662	-5.647	-5.651	-5.652
$C_{1,10}$	-4.712	-4.630	-4.596	-4.593	-4.595
$C_{1,11}$	-4.528	-4.546	-4.549	-4.554	-4.555
$C_{1,12}$	-4.604	-4.554	-4.545	-4.544	-4.546
$C_{1,13}$	-4.572	-4.558	-4.543	-4.544	-4.547
$C_{1,14}$	-4.550	-4.559	-4.550	-4.553	-4.555
$C_{1,15}$	-4.665	-4.628	-4.596	-4.592	-4.595
$C_{1,16}$	-5.658	-5.674	-5.648	-5.650	-5.652

Table 2: Convergence as a function of the expansion order in the 8+8 bus crossing example. No truncation of nearby terms.

Table 3 shows the effect of truncating nearby interaction terms as discussed in Section for various truncation parameters ε_T in equation (21). In the example shown, values up to $\varepsilon_T = 2$ result in errors below five percent.

ε_T	0.0	0.5	1	2	5
nz*10^3	18530	4186	2890	1774	725
$C_{1,1}$	82.007	82.033	82.193	82.810	85.314
$C_{1,2}$	-28.498	-28.489	-28.598	-28.919	-29.904
$C_{1,3}$	-2.340	-2.352	-2.493	-2.351	-2.530
$C_{1,4}$	-1.237	-1.237	-1.173	-1.215	-1.236
$C_{1,5}$	-0.361	-0.361	-0.353	-0.339	-0.333
$C_{1,6}$	-0.498	-0.499	-0.495	-0.507	-0.498
$C_{1,7}$	-0.319	-0.318	-0.320	-0.317	-0.329
$C_{1,8}$	-0.477	-0.479	-0.478	-0.478	-0.482
$C_{1,9}$	-5.662	-5.672	-5.652	-5.646	-5.733
$C_{1,10}$	-4.630	-4.641	-4.645	-4.722	-4.718
$C_{1,11}$	-4.546	-4.542	-4.506	-4.569	-4.993
$C_{1,12}$	-4.554	-4.556	-4.559	-4.596	-4.657
$C_{1,13}$	-4.558	-4.546	-4.578	-4.569	-4.812
$C_{1,14}$	-4.559	-4.566	-4.521	-4.656	-4.785
$C_{1,15}$	-4.628	-4.620	-4.651	-4.647	-4.816
$C_{1,16}$	-5.674	-5.687	-5.703	-5.778	-5.875

Table 3: Non-zero entries (nz) and capacitances for the 8+8 bus crossing structure when increasing the truncation parameter ε_T, $p = 2$.

Complexity

As a measure of the overall cost, the number of non-zero terms in the non-standard form are counted. The remaining steps of the algorithm, namely setting up the basis and transforming vectors into the multiscale basis, are negligible compared to setting up the matrix and performing matrix-vector multiplications.

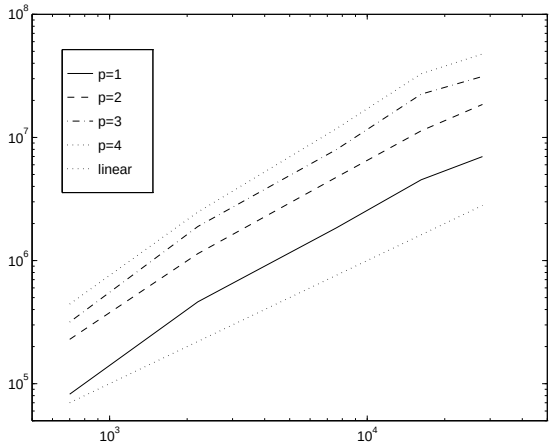

Figure 1: Complexity for ellipsoid

Figure 1 shows the number of non-zero entries in the non-

standard form as a function of the number of conductors in the bus-crossing structure for various expansion orders. These counts were obtained without truncating terms in the nearby interaction matrices described in Section . For a fixed expansion order p the complexity grows faster than linearly for the problems with a few conductors, but eventually reaches linear complexity with the number of panels. Although we have not demonstrated this here, the number of non-zero terms can be shown to be exactly equal to the number of entries in all M2L, Q2P, Q2L, and M2P matrices in the Fast Multipole algorithm. Here we use the terminology from [13]. Since the M2M and L2L operations are negligible, the cost for the calculating of the potential due to a charge distribution in the multiscale basis is almost exactly equal to the cost of the FMM. Note that in our experiments we use only first nearest neighbors. If second-nearest neighbors are included, the operation count for both methods go up by the same factor, between 1.5 to 2, depending on the problem geometry.

Different to the FMM, the wavelet basis allows truncation of nearby interactions as described in Section . The effect is a dramatic reduction of the entries in the matrix without compromising the accuracy of the approximation. This is clearly evidenced in Table 3.

Preconditioning

We compare the number of iterations of the cg algorithm with and without preconditioner. The iteration is terminated until the initial residual is reduced by at least 10^{-9}. The iteration counts in Table 4 clearly show the effectiveness of the multiscale preconditioner: When increasing the problem complexity by adding more conductors the number of iteration go up without preconditioner, however, with the preconditioner the numbers appear to remain bounded.

conductors	1+1	2+2	4+4	6+6	8+8
cg	58	79	96	110	124
pcg	12	17	18	18	18

Table 4: Iterations versus number of conductors, cg and preconditoned cg.

8 Conclusions

We described a new multiscale method for accelerating boundary-element methods for 3-D capacitance extraction and presented several computational results with comparisons to the publically available FASTCAP program. We showed that with truncation of nearby terms, which can only be used with the multilevel method, it is possible to improve the sparsification of the boundary element method by a factor of five over FASTCAP without compromising accuracy for even the smallest coupling capacitances. We also show that the multiscale preconditioner reduces the cost of solving the boundary-element equations by as much as a factor of five, and when combined with the better sparsification gives a speed improvement over FASTCAP of anywhere from 25 to 300.

9 Acknowledgements

The authors would like to thank Keith Nabors, the author of FASTCAP, for making his program publically available. This work was supported by the DARPA composite CAD program, the DARPA muri program, and grants from the Semiconductor Research Corporation and the MAFET consortium.

References

[1] W. Shi, J. Liu, N. Kakani, and T. Yu, A Fast Hierarchical Algorithm for 3-D Capacitance Extraction *Proceeding of the 29th Design Automation Conference*, San Francisco, CA, June, 1997, pp. 212-217.

[2] V. Veremey and R. Mittra, A Technique for Fast Calculation Of Capacitance Matrices of Interconnect Structures *IEEE Transactions of Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, Vol 21, No 3, pp. 241-249.

[3] W. Hong, W. K. Sun, Z. H. Zhu, H. Ji, B. Song, and W. Dai, A Novel Dimension-Reduction Technique for the Capacitance Extraction of 3-D VLSI Interconnects *MTT*, Vol. 46, No. 8, pp 1037-1044.

[4] J. R. Phillips and J. K. White, "A Precorrected-FFT method for Electrostatic Analysis of Complicated 3-D Structures," *IEEE Trans. on Computer-Aided Design*, October 1997, Vol. 16, No. 10, pp. 1059-1072.

[5] S. Kapur and J. Zhao,"A fast method of moments solver for efficient parameter extraction of MCMs" Design Automation Conference, 1997 pp. 141–146.

[6] G. Beylkin, R. Coifman, and V. Rohklin. Fast wavelet transforms and numerical algorithms. *Comm. Pure Appl. Math.*, XLIV:141–183, 1991.

[7] K. Nabors and J. White, "FastCap: A Multipole-Accelerated 3-D Capacitance Extraction Program," *IEEE Transactions on Computer-Aided Design*, vol. 10 no. 10, November 1991, p1447-1459.

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

[9] A. E. Ruehli and P. A. Brennan, "Efficient capacitance calculations for three-dimensional multiconductor systems," *MTT*, vol. 21, pp. 76–82, February 1973.

[10] Y. L. Le Coz and R. B. Iverson, "A stochastic algorithm for high speed capacitance extraction in integrated circuits," *Solid State Electronics*, vol. 35, no. 7, pp. 1005–1012, 1992.

[11] Lesslie Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, Cambridge, Massachusetts, 1988.

[12] Harry Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

[13] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory. *SIAM J. Sci. Statist. Comput.*, 15(3):713–735, 1994.