Generic approaches to parasitic extraction problems

Joel R. Phillips
Cadence Berkeley Laboratories, San Jose, CA 95134

Abstract

Integral equation methods have become popular for electromagnetic analysis problems such as computation of interconnect parasitics. However, developing integral equation codes that can treat diverse physics and interface with solvers in other domains requires algorithms that can easily be adapted to a variety of geometrical descriptions, solver interfaces, and integral equation formulations. In this paper we survey some of the most popular fast integral equation solution techniques with mind to their flexibility in dealing with diverse problem domains.

I. Introduction

Detailed analysis of electromagnetic phenomena in modern integrated circuit processes is difficult for three reasons : large scale of the problems to be analyzed, complexity of geometries encountered, and the diversity of physics that must be simultaneously analyzed.

Modern digital chip designs with millions of gates require accurate extraction of tens of millions of parasitic resistances, capacitances, and, soon, inductances. Each parasitic element represents the electrical behavior of a complex three-dimensional object. A typical approach to parasitic extraction is to characterize the electrical behavior of a typical set of objects and store the information for each object in a database. Parasitic extraction of the entire chip is performed by, for each parasitic element, trying to find a similar object, or pattern, in the database and using these pre-existing patterns to estimate the parasitics. Many such patterns must be thoroughly characterized to create the database, and this requires efficient and accurate electromagnetic analysis codes. More accurate analyses can be obtained by directly subjecting relevant portions of layout to an analysis using a high-capacity field-solver[1, 2]. This approach is particularly well suited to analysis of critical nets in a design, where accurate estimation of performance is essential.

A wide variety of numerical schemes are available for such electromagnetic analysis problems, but in the last decade, integral equation methods have attracted particular interest. The advantage of integral equation methods is that they can efficiently handle complex three-dimensional geometries with a minimal number of discretization elements, because only the most essential portions of the problem domain need be discretized. For example, for capacitance calculations, only the charge-bearing conductor surfaces must be discretized, while for inductance, only the current-carrying interior conductor volume must be represented. The disadvantage of integral equation schemes, compared with alternatives such as finite-difference or finite-element methods, is that they generate large systems of linear equations with dense matrices. The dense matrix representing a system of n simultaneous linear equations in general requires $O(n^2)$ storage, and traditional Gaussian-elimination based solution algorithms require $O(n^3)$ operations. When n is greater than a few thousand, explicit manipulation of these dense matrices becomes impractical. Much effort has been put into devising algorithms that can perform operations with the dense matrices arising in integral-equation methods in an efficient, implicit manner[3, 4, 5]. The typical approach is to concentrate on accelerating solution of the linear equations generated by discretization of integral equations.

II. Fast Integral Equation Solvers

To obtain a concrete example, consider the model problem of capacitance extraction. The capacitance of an m-conductor geometry is given by a (symmetric) matrix $C \in \mathbf{R}^{m\times m}$ that maps a set of m vectors of conductor voltages V to conductor charges Q as $Q = CV$. That is, V_{kl} represents the potential of the kth conductor, for the lth test vector, Q_{kl} is the charge on conductor k generated by the lth voltage vector, and the entry C_{kl} represents capacitive coupling between conductors l and k. To extract capacitances, the charges Q_{kl} must be determined given the voltages V. Formally this means it is necessary to solve Laplace's equation for a sequence of m Dirichlet boundary conditions on the conductor surfaces. If the conductors are embedded in an infinite homogeneous dielectric, such as free space, a first-kind integral equation may be written[6, 7, 8] for the charge density σ which lies on the conductor surfaces,

$$\psi(x) = \int_{surfaces} \sigma(x') \frac{1}{4\pi\varepsilon \|x - x'\|} da' \quad , \qquad (1)$$

where $\psi(x)$ is the known conductor surface potential, da' is the differential conductor surface area, x, $x' \in \mathbf{R}^3$, ε is the dielectric constant, and $\|x\|$ is the Euclidean length of x.

A standard approach[9] to numerically solving (1) for the charge density σ is to use a piece-wise constant collocation scheme. In this approach the conductor surfaces are approximated by a set of n polygons, or "panels", and it is assumed that on each panel i, a charge, q_i, is uniformly distributed, as in Figure 1. For each panel, an equation is written which relates the known potential at the center of that i-th panel, denoted $\overline{f}_i$ and given at the lth potential solution by $\overline{f}_i = V_{kl}$ if panel i is on conductor k, to the sum of the contributions to that potential from the n charge distributions on all n panels. The result is the dense linear system,

$$Pq = \overline{f} \qquad (2)$$

0-7803-6279-9/00/$10.00 © 2000 IEEE

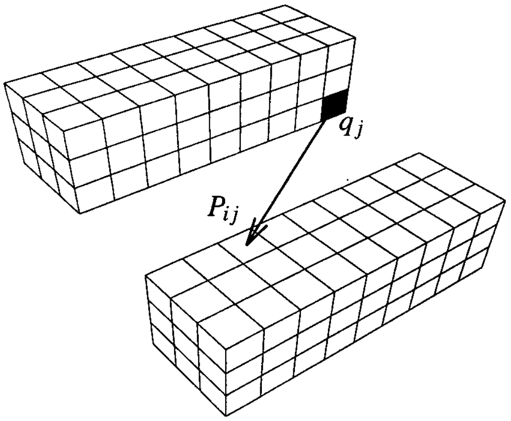

Figure 1: Piecewise-constant collocation discretization of t conductors. Conductor surfaces are discretized into panels wh support a constant charge density.

where $P \in \mathbf{R}^{n \times n}$, $q \in \mathbf{R}^n$ is the vector of panel charges, $\bar{f} \in \mathbf{R}'$ the vector of known panel potentials, and

$$P_{ij} = \frac{1}{a_j} \int_{panel_j} \frac{1}{4\pi\varepsilon ||x_i - x'||} da', \qquad (3)$$

where the collocation point x_i is the center of the i-th panel and a_j is the area of the j-th panel.

The typical approach to the problem is to solve 2 by an Krylov-subspace based iterative method such as GMRES[10]. To solve a linear system, GMRES requires only matrix vector products, that is, the operation of computing Pv given an arbitrary vector v. Nominally, because each source point contributes to each evaluation point in the integral, this is an operation with $O(n^2)$ complexity. To a large extent, research in integral equations methods over the past ten years has focused on accelerating the operation Pv relative to naive pairwise evaluation of the interactions. Many algorithms have been proposed for this acceleration step, such as fast multipole methods[11, 3], grid-based methods[12, 4], SVD approaches[5], and wavelet-like algorithms[13, 14, 15].

An emerging issue for the integral equation solvers is that they must be able to handle diverse physics. For example, in large-scale mixed-signal and – particularly – integrated RF circuits (RFICs), a variety of electromagnetic analyses must be performed to adequately characterize parasitics. Interconnect capacitances and resistances are extracted using an electroquasistatic paradigm, as previously discussed. Substrate parasitics are physically similar, but require substantially different extraction procedures to be analyzed efficiently[16]. Packaging may critical for performance, and may be analyzed with electromagnetoquasistatic [17] or full-wave [18] methods. Integrated chip passive components such as spiral inductors also require very accurate characterization, possibly full-wave analysis. Each of these problems possess different geometrical representations and possibly distinct physics. An even more challenging scenario is presented by microelectricalmechanical systems (MEMS). Here, the electromagnetic solvers must integrate with computations in a broad array of different physical domains, such as mechanical and thermal analysis. In addition, the integral-equation equation techniques developed for electromagnetics are being adapted for use on other problems, such as microfluidics [19].

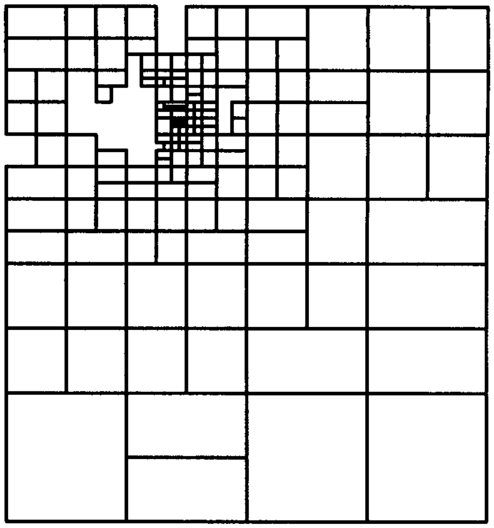

Figure 2: Portion of hierarchical binary-tree spatial decomposition for FMM-class algorithms showing multipole approximation regions as seen by a single evaluation point. Note spatial refinement near evaluation point, and empty areas where no charge sources are present. There are 157 individual cells in this example.

Problem diversity manifests itself in two ways in the integral-equation based codes. First, the physics of the problem being analyzed by the core integral equation solver itself may vary. This typically manifests itself as a change in the integral equation formulation and/or a change in the kernel of the integral equation. As a result there have been many research efforts designed to produce algorithms that are adaptable to a variety of kernels. We survey some of the most popular acceleration techniques in Sections III.-V.. In Section VI., we will summarize the methods and discuss some possibilities to dealing with the proliferation of application domains, interfaces, and integral equation formulations..

III. Fast Multipole Methods

The first fast integral equation solvers[3] were developed for the Laplace problems, as represented by the $1/r$ kernel in (1), and based on fast-multipole methods[20]. Fast multipole methods (FMMs) have been extended to a wide variety of problems[21], some of them, such as polynomial interpolation, seemingly unrelated to the $1/r$ type kernels that arise in the Laplace problems. They all share some common characteristics, however, and have motivated many of the alternative approaches discussed in Sections IV. and V.. We will include the "tree codes" [22] as members of the fast-multipole class, since as explained below, for our purposes the determining feature of this class of algorithm is the dependence on certain theorems from classical physics.

Fast multipole methods for computation of "potentials" are based on three key ideas. First, there is a notion of the separation of the "near" and "far" field for a given kernel. For a given evaluation point x in (1), the near-field contribution consists of all portions of the integral derived from sources σ at a position x' with separation $||x - x'||$ less than some value R. The far-field is due to all other sources. In a direct evaluation of the integral, it is the far-field terms that represent the majority of the evaluation cost. Efficient calculation of the far-field contributions is necessary to reduce the overall computational complexity.

The second key idea is the introduction of a hierarchical spatial decomposition. Figure 2 shows a spatial decomposition as seen from the perspective of a single potential evaluation point. Each rectangular cell represents a region within which far-field contributions to this evaluation point are collected together in order to reduce the computational cost. Because the representations of the far-field in each supercell can be used for many evaluation points, the cost of the algorithm is proportional to the number of cells needed for a single evaluation multiplied by the number of evaluations. If the number of cells in a typical spatial decomposition is much less than the number of original source points in the integral, the operation count for the potential computation can be reduced. Other techniques to further reduce the operation count, such as amortizing the cost of the far-field interaction computations over many evaluation points by use of local expansion representations[11], are available. Since the number of cells in such a spatial hierarchy grows only linearly with the number of source points, the net result is a reduction of complexity from $O(n^2)$ to $O(n)$ or $O(n\log n)$ depending on the details of the algorithm, spatial hierarchy, and type of far-field representation.

The third feature of fast multipole algorithms has to do with the details of the far-field representations. The original algorithms made use of *addition theorems* to separate the potential computation into a part that is due to source terms only, and a part due to evaluation points only. For example, the $1/r$ kernel can be written as a series expansion

$$\frac{1}{||x-x'||} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{1}{r} \left(\frac{r'}{r}\right)^{l} Y_{lm}(\theta,\phi) Y^{*}_{lm}(\theta',\phi') \qquad (4)$$

where (r,θ,ϕ) and (r',θ',ϕ') are the spherical coordinates of the points x and x' respectively. In each cell, multipole expansions of the charges can be performed by calculating the multipole moments, $m_{00} = \int \sigma(x'), m_{1m} = \int \sigma(x') r' Y^{*}_{1m}(\theta',\phi')$ and so forth. These coefficients represent an expansion of the potential due to sources in a cell in terms of $(1/r)$, where r is the distance to the expansion point, usually the cell center, and the spherical harmonics Y_{lm}. In practice, what is done is to perform multipole representations of the leaf cells in the tree, then calculate multipole representations of the higher order cells by shifting the origin of the multipole representation in the child cells to the origin of the parent cells. This procedure is applied recursively up the tree hierarchy until the top level is reached, at which time a multipole representation of the sources in each cell, at every hierarchy level, will have been computed. To evaluate the potential in a given cell, the multipole expansions can be converted into local expansions, that is, expansions of the potential outside an evaluation cell in terms of powers of r' and spherical harmonics Y_{lm}. (We can consider the special case of potential evaluation as evaluation of an order-zero local expansion.) These local expansions can likewise be passed down the spatial hierarchy to lower levels. Accuracy can be insured in the overall algorithm by requiring that evaluation of multipole expansions only occurs for cells that are well-separated, that is, for small (r'/r). Then if the series is truncated at small l,m, necessary for computational efficiency, the error induced by series truncation can be bounded at a small constant. Various modification of this approach are needed, particular for kernels with oscillatory parts such as e^{ikr}/r, but the essential features remain intact. To summarize, those features are

1. Construction of a spatial hierarchy.
2. Construction of multipole-to-multipole (M2M), multipole-to-local (M2L), and local-to-local (L2L) operators.
3. Hierarchical application of the M2M, M2L, and L2L operators for given distributions of source and evaluation points. For source/evaluation points that are very close, usually interactions within the same cell or between points in cells that are contiguous, the interactions reduce to evaluations of the original integral (1).

The difficulty with this approach occurs when one is interested in more than one kernel. Construction of the M2M, M2L, and L2L operators requires detailed analytical knowledge of the kernel under study. This makes development of FMM algorithms for kernels representing new physics a laborious process. Often, no closed-form representation of the kernel is available at all. This situation has led to a search for algorithms that can be quickly adapted to a wide variety of kernels. No truly general purpose algorithm exists, but many algorithms can be devised that are applicable to kernels in a class defined by a suitable, sufficiently general, restricting assumption.

IV. Grid Based Methods

An alternative to fast multipole methods are methods which use a grid, instead of multipole expansions, to represent the far-field. These methods exploit the fact that potentials at evaluation points distant from a cell can be accurately computed by representing the given cell's charge distribution using a small number of (fictitious) weighted point charges. Constructing such a representation is equivalent to interpolating potentials, evaluated on the grid, to the actual potential evaluation points (for example, on the discretized panels). Thus any interpolation scheme (e.g., polynomial interpolation) can be used for the representation step, and vice versa. By applying this idea in a hierarchical manner, $O(n)$ or $O(n\log n)$ multigrid algorithms for the potential computation Pv can be constructed[12]. If the point charges all lie on a uniform grid, then the critical step in the algorithm, the evaluation of the grid potentials due to the grid charges, analogous to the M2L operations of the FMM algorithms, can be simplified. Since the potential at the grid points due to the grid charges is a discrete convolution, the computation can be performed using the FFT. Since only the far-field computations are represented on the grid, an additional correction step is needed for the near-field interactions. We summarize the precorrected-FFT[4] method for approximating Pv, illustrated in Figure 3 where we have notated steps that are conceptually similar to the FMM operations of M2M, M2L, and L2L, as

1. project the panel charges onto a uniform grid of point charges, (M2M)
2. compute the grid potentials due to grid charges using an FFT, (M2L)
3. interpolate the grid potentials onto the panels (L2L), and
4. directly compute nearby interactions (M2L).

This algorithm has several advantages. Given appropriate interpolation operators (and polynomials are always available), integrals with any kernel that satisfies certain symmetry properties can be directly treated. In particular, it is not necessary to specify the kernel in closed form. This is particularly convenient when the kernel itself is only available as the result of a complicated numerical operation. To be able to exploit the FFT, the kernel must be representable

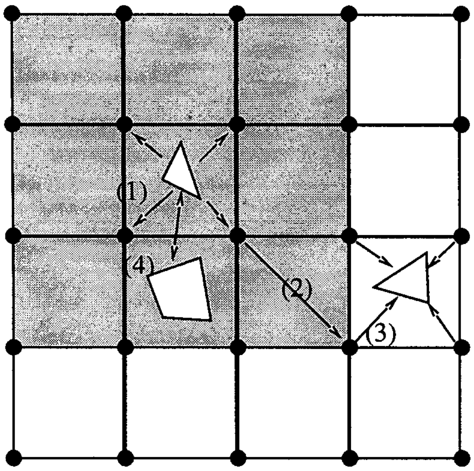

Figure 3: 2-D Pictorial representation of the four steps of the precorrected-FFT algorithm. Interactions with nearby panels (in the grey area) are computed directly, interactions between distant panels are computed using the grid.

in terms of operators that have some sort of easily accessible translational invariance. That is, it must be possible to decompose the matrix that maps grid charges to grid potentials into a set of K matrices, where K is possibly problem-dependent but independent of n, each of which has low displacement rank with respect to a matrix with Toeplitz-like structure. For example, the problem of capacitance calculation of conductors embedded in a three-layer (two-interface) dielectric medium leads to a matrix with a 2×2 block structure where each block can be expressed as a sum of block matrices, with blocks that have either Toeplitz or Hankel structure[23]. Many problems of interest, particularly in electromagnetic analysis, possess these sort of translational symmetries. Many of the problems that do not are not well suited for solution by integral equation methods in any event, because lack of translational symmetry implies that the kernel has a large number of interactions that must be calculated in a detailed pairwise manner, and therefore, the kernel may be difficult to compute and/or not amenable to acceleration techniques of any sort. In particular, we point out that this algorithm is one of the few general-purpose approaches suitable for problems with Helmholtz kernels such as e^{ikr}/r. A disadvantage of the algorithm is that it suffers from sub-optimal complexity for sufficiently inhomogeneous geometries. For practical electromagnetic problems, $O(n^{4/3})$ complexity is often the worst observed, even for oscillatory kernels[23]. In addition, for problems of moderate size, the small constant factors (compared to fast multipole algorithms) present in the algorithm often offset the higher asymptotic complexity.

V. Wavelet-Like Methods

The wavelet-like methods are motivated by the same philosophy as the FMMs, but they are constructed in such a way as to allow more problem-dependent geometrical information to be exploited in the computation. The FMMs were able to accelerate the potential computation by compressing the representation of many far-field interactions into a few multipole coefficients; only the essential information in the far-field was retained. The wavelet-like methods seek to exploit this same property, but in a less direct manner.

The wavelet-like methods achieve matrix compression by constructing a multiresolutional basis of functions with vanishing moments. The moments of a function are simply the inner products of that function with monomial terms of increasing order; they are analogous to the multipole expansion coefficients. For example in one dimension, the moments m_k of a function $h(x)$ defined on a domain S are

$$m_k = \int_S x^k h(x)dx. \tag{5}$$

If a function can be represented by an order-q polynomial then its inner product with a function known to have its first $q+1$ moments vanishing is guaranteed to be zero, regardless of the other details of the function. Thus if $\phi_0, \phi_1, \ldots, \phi_q, \psi_1, \psi_2, \ldots$ is an orthonormal basis on S, if $h(x)$ is well represented by a qth order polynomial, only the $q+1$ ϕ-terms are needed to represent $h(x)$. As basis functions with vanishing moments have a very small contribution in the far field, the wavelet-like basis produces a neat separation of near and far-field interactions that allows the dense integral operator to be represented efficiently.

Suppose a spatial hierarchy containing all the charge sources and potential evaluation points has been constructed. Consider the submatrix $P_{[ij]}$ that represents the interaction between m_i sources and m_j evaluation points in cells i and j. Suppose an orthonormal basis for cell j is given by the $m_j \times m_j$ matrix $V_j = [\Phi_j \quad \Psi_j]$ with the matrix Φ_j spanning a space of q (multidimensional) polynomials of order up to Q. Note this implies that the $m_j - q$ basis functions comprised by the columns of Ψ_j have q vanishing moments. Let Φ_i, Ψ_i be defined similarly. Then, in this basis, the potential coefficient submatrix becomes

$$\tilde{P}_{[ij]} = V_i^T P_{[ij]} V_j = \begin{bmatrix} \tilde{P}_{[11]} & \tilde{P}_{[12]} \\ \tilde{P}_{[21]} & \tilde{P}_{[22]} \end{bmatrix} = \begin{bmatrix} \Phi_i^T P_{[ij]} \Phi_j & \Phi_i^T P_{[ij]} \Psi_j \\ \Psi_i^T P_{[ij]} \Phi_j & \Psi_i^T P_{[ij]} \Psi_j \end{bmatrix} \tag{6}$$

Now, if i and j are well-separated cells, then the interaction between them can be accurately described with low-order polynomial interpolation, and thus the terms $\tilde{P}_{[12]}, \tilde{P}_{[21]}$ and $\tilde{P}_{[22]}$ will vanish to within the accuracy of the interpolation. The term $\tilde{P}_{[11]}$ will be dealt with at higher levels of the spatial hierarchy. The Φ functions from several child cells will be combined together, and a new orthonormal basis constructed with Φ' and Ψ' functions spanning and vanishing over a polynomial basis at the next higher level of the hierarchy. This process continues to the highest level of the hierarchy, where only cells containing basis functions Φ remain. In this multiresolutional basis, the integral operator will be sparse because only nearby cells will have significant interactions. Figure 4 shows the numerical sparsity pattern typical of an integral operator represented in such a multilevel basis.

Now let us consider finding the multiresolutional basis. Let M be a matrix that maps a vector representing a function f into the moments of f. If $Mf = 0$, then f has vanishing moments, or in other words, if f has vanishing moments, it must lie in the nullspace of M. Thus we can compute the basis at each leaf cell by constructing the moment matrix M and taking the singular value decomposition $M = USV^T$ where U and V both are matrices with orthonormal columns. It is more instructive to write this in the form

$$M = [U] \left[\, S_r \mid 0 \, \right] \begin{bmatrix} \Phi^T \\ \Psi^T \end{bmatrix} \tag{7}$$

where

$$S = \left[\, S_r \mid 0 \, \right] \quad V = \left[\, \Phi \quad \Psi \, \right]. \tag{8}$$

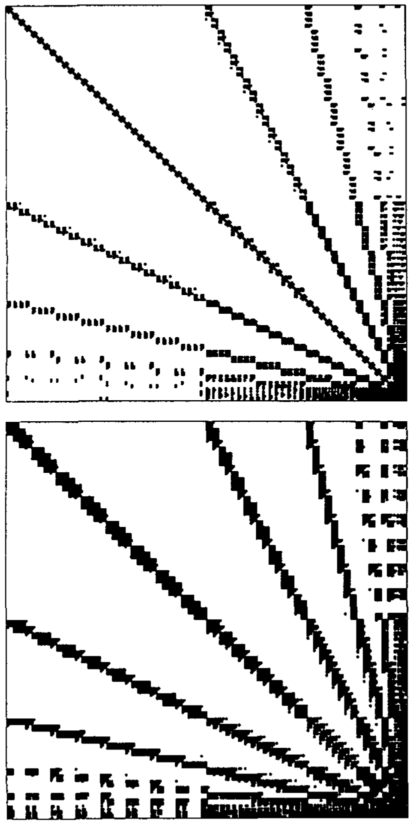

Figure 4: Top: Numerical sparsity pattern of an integral operator in a multi-level basis. Bottom: Numerical sparsity pattern of its inverse.

By construction the rightmost $m_i - q$ columns of the singular value matrix S are 0 and so then the $m_i - q$ columns of the submatrix Ψ are the columns of the matrix V corresponding to the basis functions with vanishing moments, as only vectors in the space spanned by the columns of Ψ can have non-zero inner product with Ψ. Similarly the q columns of the submatrix Φ give the basis functions with non-vanishing moments.

The basis functions Φ will be "pushed up" to the next level in order to construct the basis functions at that level. At that level, the moments of the parent cell will be needed. These can be obtained from the vector of child moments and an *M2M* type operator. The child moments can be obtained in the new basis by re-organizing the decomposition above as $MV = US$; US is clearly an operator that maps a function in the new basis to moments.

The multiresolutional approach is fairly general. Use of the polynomial moments to construct the basis allows any operator that is asymptotically smooth, i.e., interpolated by a polynomial at many length scales, to be efficiently represented. In addition, when more efficient interpolation information is available, such as for the $1/r$ kernel, it may be exploited to produce an even more compact representation. Any suitable approximation basis, in particular multipole moments[15] and M2M translation operators, can easily replace the polynomial moment operators described above without substantially altering the computational structure of the algorithm. The main drawback of this approach is that it does not appear to be suitable for oscillatory kernels such as e^{ikr}/r.

The wavelet-like basis can even be used to construct a sparse representation of an operator, and thus accelerate computations, when the operator itself is only given in an implicit form. In [16], a wavelet-like basis was used to accelerate computation of the matrix describing parasitic resistive interactions due to substrate coupling. In this case, the integral operator is only obtained by manipulating the inverse of a matrix operator. In the application of [16], the inverse operator was actually obtained via a finite-difference based discretization of the substrate. Figure 4 shows an example of the sparsity structure obtained when inverting a sample integral operator. The characteristic multiresolutional sparsity structure is still evident, but compared to the smooth forward operator, for a given numerical threshold used to drop elements in the operator, the inverse operator is somewhat less sparse.

VI. Conclusion

In this paper we have attempted to evaluate fast potential evaluation methods suitable for forming the core of large-scale integral equation solvers according to the restricting assumptions that define the class of problems for which the algorithms are applicable.

The FMM class of algorithms is characterized by a common, general computational structure (spatial trees). The general operation of the algorithms can be specified relatively independently of the details of the spatial decomposition (e.g., two vs. three dimensions, oct-trees vs. binary trees) and the details of the M2M, M2L, and L2L operators. However, deriving these operators generally is fairly troublesome, and so these methods are most useful for common kernels like $1/r$ and e^{ikr}/r. For problems where the kernels has such a known analytic structure, and a fairly inhomogeneous geometry, the FMM algorithms are worth consideration particularly when high accuracy is required. However, if the kernel is non-oscillatory, the wavelet-like methods seem to have most of the advantages of FMMs with the possibility for higher degrees of numerical compression[15].

The grid-based algorithms also have a very general, but rigid, computational structure. Within the constraints of spatial invariance, broadly defined, the precorrected-FFT method is suited to a wide variety of problems. It is fast for low accuracy calculations and has minimal setup overhead. For problems that are not strongly geometrically inhomogenous, with complicated kernels that may have oscillatory components, this algorithm is the best choice.

The wavelet-like algorithms appear to be particularly promising general purpose methods. They are the only methods that appear to be well suited for the representation of implicit and/or inverse operators. They also seem to be efficient, generating good compression ratios, and when analytic information about the kernel is available, such as the more efficient multipole representations for $1/r$ kernels, it can be incorporated into the multilevel scheme to increase its efficiency. The one drawback is that, so far, it is unclear how to efficiently represent oscillatory kernels in this sort of general framework.

An important method we have not discussed in detail is the SVD/interpolation based approach of [5]. For our purpose, the advantages and shortcomings of this method are similar to the wavelet-based approaches. The only real restriction is that the long-range interactions satisfy an interpolation condition. General kernels can be treated easily, but the method is not well suited for oscillatory kernels.

At this point, therefore, we can conclude that there is no single approach that is best for all problems likely to be encountered in integrated circuit analysis problems. Each of the approaches discussed here in detail (FMM, precorrected-FFT, and wavelet-like algorithms) have substantial advantages over the others in some important problem domain. With no clear best algorithm, the task of constructing generic platforms to address diverse applications might seem problematic. However, although the algorithms are quite diverse mathematically, there are strong similarities between the essential computational and algorithmic structures of the algorithms. The FMM, wavelet, and SVD classes of algorithms are particularly closely related.

Generic programming methodologies have recently been shown to have substantial advantages for numerical computing applications, offering the potential to work at high levels of abstraction, with generic software components, while at the same time retaining or even increasing numerical efficiency[24]. A reasonable path forward therefore seems to be to attempt to identify the least common algorithmic subsets of the computational structure of the major algorithms discussed above, and then instantiate that knowledge in an extensible, adaptable software platform. On a given problem domain, rapid comparison of the available approaches could be made and the best for a given application selected.

References

[1] M. Bachtold, M. Spasojevic, C. Lage, and P. B. Ljung, "A system for full-chip and critical net parasitic extraction for ULSI interconnets using a fast 3D field solver," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 325–338, March 2000.

[2] S. Kapur and D. Long, "Large scale capacitance extraction," in *Proceedings 37th Design Automation Conference*, Los Angeles, California, June 2000, pp. 744–749.

[3] 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*, vol. 10, no. 11, pp. 1447–1459, November 1991.

[4] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3D structures," *IEEE Trans. CAD*, pp. 1059–1072, 1997.

[5] S. Kapur and D. Long, "IES^3: a fast integral equation solver for efficient 3-D extraction," in *Proc. IEEE International Conference on Computer Aided Design*, November 1997, pp. 448–455.

[6] A. E. Ruehli and P. A. Brennan, "Efficient capacitance calculations for three-dimensional multiconductor systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 21, no. 2, pp. 76–82, February 1973.

[7] 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*, vol. MTT-32, no. 11, pp. 1441–1448, November 1984.

[8] D. Colton and R. Kress, *Integral Equations Methods in Scattering Theory*, Krieger, Malabar, Florida, 1992.

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

[10] Youcef Saad and Martin Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, no. 3, pp. 856–869, July 1986.

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

[12] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernals," *Computer Physics Communications*, vol. 65, pp. 24–38, 1991.

[13] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms," *Communications on Pure and Applied Mathematics*, vol. 44, pp. 141–183, 1991.

[14] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, "Wavelet-like bases for the fast solution of second-kind integral equations," *SIAM J. Sci. Comput.*, vol. 14, no. 1, pp. 159–184, 1993.

[15] J. Tausch and J. White, "A multiscale method for fast capacitance extraction," in *Proceedings of the 36th Design Automation Conference*, New Orleans, LA, 1999, pp. 537–542.

[16] J. Kanapka, J. Phillips, and J. White, "Fast methods for extraction and sparsification of substrate coupling," in *Proc. 37th Design Automation Conference*, June 2000, pp. 738–743.

[17] M. Kamon, N. Marques, and J. White, "FastPep:a fast parasitic extraction program for complex three-dimensional geometries," in *International Conference on Computer Aided-Design*, November 1997, pp. 456–460.

[18] J. R. Phillips, E. Chiprout, and D. D. Ling, "Efficient full-wave electromagnetic analysis via model-order reduction of fast integral transforms," in *Proceedings 33rd Design Automation Conference*, Las Vegas, Nevada, June 1996, pp. 377–382.

[19] N. Aluru and J. White, "A fast integral equation technique for analysis of microflow sensors based on drag force calculations," in *Proc. MSM 98*, April 1998, pp. 283–286.

[20] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comp. Phys.*, vol. 73, pp. 325–348, December 1987.

[21] L. Greengard, "Fast algorithms for classical physics," *Science*, vol. 265, pp. 909–914, 1994.

[22] A. W. Appel, *SIAM J. Sci. Statist. Comput.*, vol. 6, pp. 85, 1985.

[23] J. R. Phillips, *Rapid solution of potential integral equations in complicated 3-dimensional geometries*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1997.

[24] J.G. Siek and A. Lumsdaine, "The matrix template library : generic components for scientific computing," *Computing in Science and Engineering*, vol. 1, no. 6, pp. 70–71, November-December 1999.