Algorithms in FastImp: A Fast and Wide-Band Impedance Extraction Program for Complicated 3-D Geometries

Zhenhai Zhu, *Member, IEEE*, Ben Song, and Jacob K. White, *Member, IEEE*

***Abstract*—In this paper, we describe the algorithms used in FastImp, a program for accurate analysis of wide-band electromagnetic effects in very complicated geometries of conductors. The program is based on a recently developed surface integral formulation and a precorrected fast Fourier transform (FFT) accelerated iterative method, but includes a new piecewise quadrature panel integration scheme, a new scaling and preconditioning technique as well as a generalized grid interpolation and projection strategy. Computational results are given on a variety of integrated circuit interconnect structures to demonstrate that FastImp is robust and can accurately analyze very complicated geometries of conductors.**

***Index Terms*—Fast integral equation solver, panel integration, parasitic extraction, preconditioner, surface integral formulation, wide-band analysis.**

I. Introduction

COLLOCATING sensitive analog circuits and rapidly switching digital logic on a single integrated circuit, as is typical in mixed signal designs, can create coupling problems that are very difficult to find and eliminate. The difficulty is that these coupling problems are often caused by simultaneous interactions between a large number of conductors. In order to help designers find these problems, there has been renewed emphasis on developing electromagnetic analysis tools capable of wide-band analysis of very complicated geometries of conductors.

In the area of electromagnetic analysis of complicated geometries of interconnect, most of the recently developed programs have been based on combining discretized integral formulations with accelerated iterative methods [1]–[7]. Though these programs and techniques have been very effective, there is still no accelerated integral equation program capable of solving full Maxwell's equations in general three-dimensional (3-D) structures with lossy conductors which is accurate from zero frequency to microwave frequencies.

Many integral formulations have been developed and can be generally categorized into four kinds according to the state variables used in these formulations. 1) Formulations using the field variables E and H have been used for decades to solve the radiation and scattering problems [8]–[11] as well as eddy current problems [12], [13]. The well-known formulations include the electric field integral equation (EFIE) and magnetic field integral equation (MFIE) [9], [14], which are also known as Stratton–Chu's formulation [15], as well as the so-called PMCHW formulation [16]–[18]. 2) Formulations using the current and charge as state variables, such as the mixed potential integral equation (MPIE) formulation [8], [19]–[22]. 3) Formulations using vector and scalar potentials as state variables; these formulations are very commonly used for solving eddy current problems [23]. 4) Formulations using virtual sources, such as virtual current or charge, are also commonly used for solving eddy current problems [24], [25].

It is well known that the EFIE formulation is not guaranteed to produce a unique solution at interior resonant frequencies for closed structures [9], [26]. Many remedies have been proposed [27]. But there still remain many unsolved problems.

The MPIE formulation has been extensively used for the analysis of microstrip structures [19]–[22] and for arbitrary shaped conductors with only surface current [28]. It was recognized in [29] that MPIE has accuracy problem at low frequencies. The so-called loop/star and loop/tree basis functions were used to overcome this low-frequency problem [29]–[31]. The MPIE formulation has also been used for the analysis of interconnects in VLSI or analog circuits. In this case, it is also known as the partial equivalent element circuit (PEEC) method [32]. Results of the MQS analysis in [2] and the EMQS analysis in [33] have clearly demonstrated that the PEEC method can produce accurate results across a wide frequency range, from zero to hundreds of gigahertz. However, unlike the microstrip structures, which are usually approximated by zero-thickness perfect or lossy conductors [19]–[22], typical interconnect structures are lossy conductors with finite thickness. Because of the skin effect, analyzing them involves a frequency-dependent discretization of the interior of conductors and the substrate ground. At high frequencies, this kind of discretization usually renders the number of piecewise constant basis functions (also called filaments) to be prohibitively large [34]. Recently, an entire-domain basis scheme has shown some promise to remedy the sit-

Manuscript received August 23, 2003; revised January 6, 2004, and May 15, 2004. This work was supported by the Semiconductor Research Corporation, the Marco Interconnect Focus Center, the Defense Advanced Research Projects Agency (DARPA) NeoCAD program managed by the Sensors Directorate of the Air Force Laboratory, the U.S. Air Force, Wright-Patterson AFB, as well as by grants from the National Science Foundation and Intel Corporation. This paper was recommended by Associate Editor C.-J. R. Shi.

Z. Zhu was with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is now with is with Cadence Berkeley Laboratories, Berkeley, CA 94704 USA (e-mail: zhzhu@alum.mit.edu).

B. Song is with Cadence Design Systems, Inc., San Jose, CA 95134 USA (e-mail: bsong@cadence.com).

J. K. White is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: white@mit.edu).

Digital Object Identifier 10.1109/TCAD.2005.847897

uation [35], but we have yet to see that it will eventually lead to a wide-band fast Maxwell's equation solver for general 3-D structures.

Another approach to address this issue is to use the surface impedance to avoid solving 3-D governing equations numerically in the interior of conductors. The surface impedance is either derived from one-dimensional (1-D) model [36] or obtained numerically by solving a two-dimensional (2-D) problem on representative cross-sections at various locations in the problem domain [37]. The surface impedance is then combined with a standard formulation such as MPIE to solve a 3-D problem [38], [39]. Although this approach is a reasonable approximation for regular conductors at medium frequencies, the surface impedance does not include 3-D effects and may not be robust at low frequencies.

The motivation behind this paper is to find a numerically stable surface integral formulation, as such formulations avoid a frequency-dependent discretization of the interior of conductors and the substrate. The formulation should be capable of wide-band analysis of arbitrary 3-D structures and it should also be easily accelerated by the well-established techniques, such as the fast multipole method [40], [41] and the precorrected fast Fourier transform (FFT) algorithm [3].

One recently developed surface integral formulation has shown promise [42], [43], but was plagued with numerical difficulties of poorly understood origin. It was shown in [44] that one of that formulation's difficulties was related to inaccuracy in the approach to evaluate integrals over discretization panels, and a more accurate approach based on an adapted piecewise quadrature scheme was proposed. In this paper, it is also shown that the condition number of the system matrix could be very large if the feature size of the structure is small, and a scaling technique is proposed to reduce the condition number. In addition, a different preconditioner than the one used in [42] is proposed to improve the memory efficiency. With these issues being resolved, the formulation is indeed valid across wide frequency range and for all feature sizes. Now the formulation is acceleration-ready.

The fast multiple method (FMM) [40], [41] has been used successfully in many applications, such as electrostatic analysis in FastCap [1] and others [5], magneto-quasistatic analysis in FastHenry [2], and full-wave analysis in the Fast Illinois Solver Code [7]. Though the algorithm is rather general, its most efficient variants are kernel-dependent. On the other hand, the precorrected FFT (pFFT) algorithm [45], which has been successfully used in many applications [3], [46], [47], is nearly kernel-independent but can be quite inefficient for highly inhomogeneous problems. Since our surface integral formulation has a number of different kernels and the problem geometry is near-planar and densely packed, the pFFT algorithm seems better suited to our formulation. In addition, as a by-product of our work, we also developed a flexible and standalone fast integral equation solver using extensions of several of the sparse matrix based ideas in [48]. Combining the fast solver with the improved surface integral formulation, we have developed a fast impedance extraction program, FastImp. Experiments using several large examples show that FastImp can perform full 3-D electromagnetic analysis of interconnect structures with millions of unknowns from zero frequency to hundreds of gigahertz.

In Section II, we will derive the surface integral formulation. In Section III, we will show how the piecewise quadrature scheme improves the accuracy of panel integration and that it solves the low frequency problem in [43]. A simple scaling and a local preconditioning technique are used in Section IV to improve the accuracy and memory efficiency of the surface integral formulation. In Section V, we will explain the extensions needed to use the pFFT algorithm to accelerate the complicated integral operators in our surface formulation. Numerical experiments are used in Section VI to demonstrate the accuracy, speed and capacity of FastImp. Finally, Section VII concludes the paper.

II. Derivation of the Surface Integral Formulation

We focus on the 3-D interconnect structures embedded in an isotropic and homogeneous medium in this paper. We assume that each conductor (denoted by $V_i, i = 1, 2, \ldots, n$) is piecewise homogeneous and the medium (denoted by V_0) surrounding these conductors is homogeneous.

A. Governing Equations

For domains of constant permittivity and permeability, the independent Maxwell's equations in time-harmonic form are [49]

$$\nabla \times \vec{E} = -j\omega\mu\vec{H} \tag{1}$$

$$\nabla \times \vec{H} = \vec{J} + j\omega\epsilon\vec{E} \tag{2}$$

$$\nabla \cdot \vec{J} = -j\omega\rho \tag{3}$$

where $\vec{E}$ is the electric field, $\vec{H}$ is the magnetic field, $\vec{J}$ is the volume current density, ρ is the net charge density, and μ and ϵ are the permeability and permittivity, respectively. The constitutive equation for conductors is

$$\vec{J} = \sigma\vec{E} \tag{4}$$

where σ is the conductivity. Equations (1) and (2) imply

$$\nabla \times \nabla \times \vec{E} - \omega^2\epsilon\mu\vec{E} = -j\omega\mu\vec{J}. \tag{5}$$

Obviously, (1)–(4) are equivalent to (1) and (3)–(5). In view of (2) and (4), we have

$$\nabla \cdot \nabla \times \vec{H} = (\sigma + j\omega\epsilon)\nabla \cdot \vec{E} = 0$$

where we have assumed homogeneity of σ and ϵ inside each conductor. Thus

$$\nabla \cdot \vec{E}(\vec{r}) = 0, \quad \vec{r} \in V_i \tag{6}$$

where $\vec{r}$ is a point in the interior of conductor V_i. This means the net charge inside a homogeneous conductor is zero [50]. Hence, (5) can be reduced to

$$(\nabla^2 + \omega^2\epsilon\mu)\vec{E}(\vec{r}) = j\omega\mu\vec{J}(\vec{r}), \quad \vec{r} \in V_i. \tag{7}$$

It should be noted that the permitivitty and permeability inside a conductor are assumed to be the same as those of the free space [51].

Equations (1), (4), (6), and (7) are the governing equations inside each conductor V_i, and (1)–(4) are the governing equations in the homogeneous medium V_0.

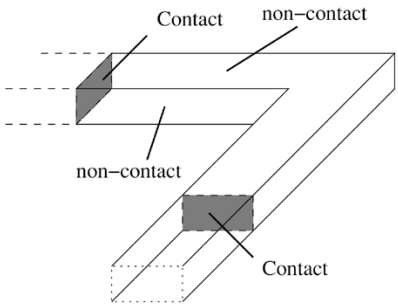


Fig. 1. Surface of a 3-D interconnect conductor.

B. Boundary Conditions

The surface of each conductor could be divided into two parts: contact surfaces and noncontact surfaces, as shown in Fig. 1. The contact is an artificially exposed surface. It is created primarily because we want to use the divide-and-conquer strategy to separate a block of 3-D interconnect from other parts within a large chip. In this paper, we use voltage source connected to the contacts as excitation and compute current through the contacts, from which the impedance can be easily calculated [32], [33], [52]. In this case, it is reasonable to assume that the total current through voltage source and the corresponding contacts is equal. Hence, there should be no charge accumulation on the contacts. So (6) also holds true on the contacts.

Because of the nature of the commonly used strategy to decompose a large chip into many smaller blocks, the conductors connected to these contacts are usually long and thin signal lines. Hence, it is reasonable to assume that the current goes into these contacts does not have the transversal components, i.e., $\hat{t}\cdot\vec{J}=0$, where $\hat{t}$ is the unit tangential vector on the contacts. Using the constitutive equation in (4) implies

$$\hat{t}(\vec{r})\cdot\vec{E}(\vec{r})=\vec{E}_t(\vec{r})=0 \tag{8}$$

if $\vec{r}$ is on a contact. Equations (6) and (8) imply that if $\vec{r}$ is on a contact

$$\hat{n}(\vec{r})\cdot\frac{\partial\vec{E}(\vec{r})}{\partial n(\vec{r})}=\frac{\partial E_n(\vec{r})}{\partial n(\vec{r})}=0 \tag{9}$$

where $\hat{n}(\vec{r})$ is the normal unit vector on the contact. Since (8) and (9) also imply (6), we will use (6) on the noncontacts only to avoid redundancy. On the other hand, since charge on a noncontact surface is not necessarily zero, in view of (3) and (4), the boundary condition when $\vec{r}$ is on a noncontact surface becomes [49]

$$\hat{n}(\vec{r})\cdot\vec{E}(\vec{r})=E_n(\vec{r})=\frac{j\omega\rho(\vec{r})}{\sigma} \tag{10}$$

where ρ is the surface charge density since the net charge inside a conductor is zero [50]. It should be noted that the position vector $\vec{r}$ in $\vec{E}(\vec{r})$ of (8) and (10) and in $(\partial\vec{E}(\vec{r}))/(\partial n)$ of (9)is only on the inner side of the conductor surfaces. It should also be noted that throughout this paper $\vec{J}$ is always the actual volume current density.

C. Surface Integral Representation

Green's second identity can be used to construct a surface integral representation of the solution to (7) [14]

$$T\vec{E}(\vec{r})=\int_{S_i}dS'\left(G_0(\vec{r},\vec{r}')\frac{\partial\vec{E}(\vec{r}')}{\partial n(\vec{r}')}-\frac{\partial G_0(\vec{r},\vec{r}')}{\partial n(\vec{r}')}\vec{E}(\vec{r}')\right)-j\omega\mu\int_{V_i}dV'G_0(\vec{r},\vec{r}')\vec{J}(\vec{r}') \tag{11}$$

where

$$G_0(\vec{r},\vec{r}')=\frac{e^{jk_0|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|},\quad k_0=\omega\sqrt{\epsilon\mu} \tag{12}$$

$$T=\begin{cases}1, & \text{if } \vec{r}\in V_i\\ 1/2, & \text{if } \vec{r}\in S_i\\ 0, & \text{otherwise}\end{cases} \tag{13}$$

and S_i is the surface of conductor V_i. When $\vec{r}\in S_i$, the surface integral in (11) should be the principal value integral [53]. From (11) and (13), one can see that the integral representation in the right-hand side of (11) generates zero field when $\vec{r}$ is outside of V_i. If we write (11) for each conductor separately but let $\vec{r}$ be fixed on the inner side of the surface of a particular conductor V_k, and then sum these equations, we obtain

$$\frac{1}{2}\vec{E}(\vec{r})=\int_{S}dS'\left(G_0(\vec{r},\vec{r}')\frac{\partial\vec{E}(\vec{r}')}{\partial n(\vec{r}')}-\frac{\partial G_0(\vec{r},\vec{r}')}{\partial n(\vec{r}')}\vec{E}(\vec{r}')\right)-j\omega\mu\int_{V}dV'G_0(\vec{r},\vec{r}')\vec{J}(\vec{r}'),\quad \vec{r}\in S_k \tag{14}$$

where $k=1,2,\ldots,n$, S is the union of all conductor surfaces and V is the union of all conductor regions.

Substituting (4) into (7) yields

$$\nabla^2\vec{E}(\vec{r})+(\omega^2\epsilon\mu-j\omega\mu\sigma_i)\vec{E}(\vec{r})=0,\quad \vec{r}\in V_i \tag{15}$$

where σ_i is the conductivity of the conductor V_i. Again, Green's second identity yields the surface integral representation of the solution to (15)

$$\frac{1}{2}\vec{E}(\vec{r})=\int_{S_i}dS'\left(G_1(\vec{r},\vec{r}')\frac{\partial\vec{E}(\vec{r}')}{\partial n(\vec{r}')}-\frac{\partial G_1(\vec{r},\vec{r}')}{\partial n(\vec{r}')}\vec{E}(\vec{r}')\right),\quad \vec{r}\in S_i \tag{16}$$

where

$$G_1(\vec{r},\vec{r}')=\frac{e^{jk_1|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|},\quad k_1=-\sqrt{\omega^2\epsilon\mu-j\omega\mu\sigma_i}. \tag{17}$$

Since (14) and (16) are the formal solutions to the same equation in slightly different forms, they are obviously equivalent. We use both to simplify the derivation.

So far, only the formal solutions to (7) inside each conductor have been found. To find the formal solution to the governing equations in region V_0, the homogeneous medium, we turn to the MPIE. Now each conductor is treated as a volume current source. In the standard MPIE formulation [8], the electric field everywhere, including the interior of every conductor, is

$$\vec{E}(\vec{r})=-j\omega\vec{A}-\nabla\phi(\vec{r})=-j\omega\mu\int_{V}dV'G_0(\vec{r},\vec{r}')\vec{J}(\vec{r}')-\nabla\phi(\vec{r}) \tag{18}$$

where

$$\phi(\vec{r}) = \int_S dS' \frac{\rho(\vec{r}')}{\epsilon} G_0(\vec{r}, \vec{r}'). \tag{19}$$

Notice that the volume integral term in (18) is identical to the one in (14), we could use this fact to eliminate this undesirable volume integral term. Letting $\vec{r} \in S_k$ in (18) and subtract it from (14), we then obtain

$$-\frac{1}{2}\vec{E}(\vec{r}) = \int_S dS' \left(G_0(\vec{r}, \vec{r}') \frac{\partial \vec{E}(\vec{r}')}{\partial n(\vec{r}')} - \frac{\partial G_0(\vec{r}, \vec{r}')}{\partial n(\vec{r}')} \vec{E}(\vec{r}') \right) + \nabla \phi(\vec{r}), \quad \vec{r} \in S_k \tag{20}$$

where $k = 1, 2, \ldots, n$. The integral representation (20) is no longer the formal solution to (7), hence it is no longer equivalent to (16). Now we have found the surface integral representation of the formal solutions to the governing equations inside conductors and homogeneous medium. It should be noted that the surface integrals in (14), (16), and (20) are all principal value integrals.

Unlike the standard MPIE, the Lorentz gauge $\nabla \cdot \vec{A} + j\omega\epsilon\mu\phi = 0$ is not explicitly enforced in our formulation because it is implied by (18), (19), and (6), which are explicitly enforced. Now it is clear that had (5), instead of (6) and (7), been used as the governing equations, we would have to enforce Lorentz gauge. This would introduce the vector potential $\vec{A}$ and ultimately a volume integral term into our formulation. Since this volume term is different from the ones in (14) and (18), it may not be possible to eliminate this undesirable term using the same trick used to derive (20).

It is worth mentioning that (11) and (16) are very similar to the standard EFIE formulation [9]. There are a few equivalent forms of EFIE; the one closest to (16) is

$$\frac{1}{2}\vec{E}(\vec{r}) = \int_{S_i} dS' \left(G_1(\vec{r}, \vec{r}') \frac{\partial \vec{E}(\vec{r}')}{\partial n(\vec{r}')} - \frac{\partial G_1(\vec{r}, \vec{r}')}{\partial n(\vec{r}')} \vec{E}(\vec{r}') \right) + \int_{S_i} dS' \left(\hat{n}(\vec{r}') G_1(\vec{r}, \vec{r}') (\nabla' \cdot \vec{E}(\vec{r}')) \right), \quad \vec{r} \in S_i. \tag{21}$$

The EFIE equation closest to (11) is (21) with the addition of a volume integral term like the one in (11), with G_1 being replaced by G_0.

The standard EFIE is derived from the vector Helmholtz equation (5) using the Green's second identity in vector form, with (6) not explicitly enforced. However, as discussed before, (6) must be enforced in our formulation. This is why we have chosen (16) rather than (21), the standard EFIE.

D. Surface Formulation

We follow the convention in the PEEC model, using the difference between ϕ on two contacts of the same conductor as the voltage excitation term [2], [33]. In light of this, we introduce (29), the last equation, into our formulation.

The boundary conditions and the surface integral representation of the solution to the Maxwell's equations are summarized as the following:

$$\frac{1}{2}\vec{E}(\vec{r}) = \int_{S_i} dS' \left(G_1(\vec{r}, \vec{r}') \frac{\partial \vec{E}(\vec{r}')}{\partial n(\vec{r}')} - \frac{\partial G_1(\vec{r}, \vec{r}')}{\partial n(\vec{r}')} \vec{E}(\vec{r}') \right), \quad \vec{r} \in S_i \tag{22}$$

$$-\frac{1}{2}\vec{E}(\vec{r}) = \int_S dS' \left(G_0(\vec{r}, \vec{r}') \frac{\partial \vec{E}(\vec{r}')}{\partial n(\vec{r}')} - \frac{\partial G_0(\vec{r}, \vec{r}')}{\partial n(\vec{r}')} \vec{E}(\vec{r}') \right) + \nabla \phi(\vec{r}), \quad \vec{r} \in S_{\rm nc} \tag{23}$$

$$\phi(\vec{r}) = \int_S dS' \frac{\rho(\vec{r}')}{\epsilon} G_0(\vec{r}, \vec{r}'), \quad \vec{r} \in S \tag{24}$$

$$\nabla \cdot \vec{E}(\vec{r}) = 0, \quad \vec{r} \in S_{\rm nc} \tag{25}$$

$$E_n(\vec{r}) = \frac{j\omega\rho(\vec{r})}{\sigma}, \quad \vec{r} \in S_{\rm nc} \tag{26}$$

$$\hat{t}(\vec{r}) \cdot \vec{E}(\vec{r}) = 0, \quad \vec{r} \in S_c \tag{27}$$

$$\frac{\partial E_n(\vec{r})}{\partial n(\vec{r})} = 0, \quad \vec{r} \in S_c \tag{28}$$

$$\phi(\vec{r}) = \text{constant}, \quad \vec{r} \in S_c \tag{29}$$

where $S_{\rm nc}$ and S_c are the noncontact part and the contact part of conductor surface S, respectively. There are eight scalar state variables, $E_x, E_y, E_z, (\partial Ex)/(\partial n), (\partial Ey)/(\partial n), (\partial Ez)/(\partial n), \phi$, and ρ. All components of $\vec{E}$ and $(\partial \vec{E})/(\partial n)$ are defined only on the inner side of the conductor surfaces. The potential ϕ and the surface charge density ρ are continuous across the surface, so no differentiation between the inner and the outer side of the surface is needed for ϕ and ρ. It should be noted that (25) is essentially same as (6) because it is enforced on the inner side of the noncontact conductor surface, which is still in the interior of the conductors.

Equations (22)–(29) form a system of nine equations which involve unknowns on conductor surfaces: three scalar equations in (22), three scalar equations in (23) complemented by two scalar equations in (27), one scalar equation in (24), one scalar equation in (25) complemented by (29), and one scalar equation in (26) complemented by (28). Since there are only eight variables and nine equations, (22)–(29) may not have a solution. In [54], it was shown that the domain truncation shown in Fig. 1 combined with artificial boundary conditions (27)–(29) insures that there is no solution. The PEEC formulation has a similar problem and so (19) was not enforced for the interior of the conductor [52]. In our formulation, we discard one of the three scalar equations in (23).

In the local coordinate system $(\hat{t}_1, \hat{t}_2, \hat{n})$, where $\hat{t}_1$ and $\hat{t}_2$ are two orthogonal unit vectors tangential to the surface of conductor and $\hat{n}$ is the unit vector normal to the surface of conductor, the term $\nabla\phi$ in (23) can be written as $\hat{t}_1(\partial\phi)/(\partial t_1) + \hat{t}_2(\partial\phi)/(\partial t_2) + \hat{n}(\partial\phi)/(\partial n)$. Using ϕ on the surface, finite differences can be used to compute $(\partial\phi)/(\partial t_1)$ and $(\partial\phi)/(\partial t_2)$, but not $(\partial\phi)/(\partial n)$. In light of the above observation, we have decided to enforce (23) only along two tangential directions

in the local coordinate system and not to enforce it along the normal direction. Equation (23) then becomes

$$-\hat{t}(\vec{r})\cdot\frac{1}{2}\vec{E}(\vec{r}) = \hat{t}(\vec{r})\cdot\int_S dS'\left(G_0(\vec{r},\vec{r}')\frac{\partial\vec{E}(\vec{r}')}{\partial n(\vec{r}')} - \frac{\partial G_0(\vec{r},\vec{r}')}{\partial n(\vec{r}')}\vec{E}(\vec{r}')\right) + \hat{t}(\vec{r})\cdot\nabla\phi(\vec{r}), \quad \vec{r}\in S_{\rm nc}. \tag{30}$$

This results in a system of eight scalar equations.

In summary, the surface integral formulation used in this paper consists of (22) and (24)–(30). For the EMQS analysis, k_0 in (12) becomes zero and the term $\omega^2\epsilon\mu$ in (17) should be dropped [55]. But the number of state variables is unchanged. For the MQS analysis, on top of above simplification, the surface charge density ρ in (26) is assumed to be zero [55]. Hence, it becomes redundant and is not used as a state variable, and (24) is eliminated. Hence, the total number of scalar unknowns and equations for MQS analysis becomes seven. It should be noted that the two slightly different sets of equations, (22) and (24)–(30) for EMQS and full-wave analysis, and (22) and (25)–(30) for MQS analysis, are all widebanded, as will be shown in the numerical result section. In addition, our numerical experiments show that they all produce virtually identical impedance at very low frequencies but they behave very differently at high frequencies. These are consistent with electromagnetic theory. One does not need to switch between these different sets of equations to achieve wide-band behavior.

E. Discretization of the Formulation

In order to discretize the integral (22), (30), and (24), a piecewise constant centroid collocation scheme is used. The conductor surface is discretized into many flat panels. Seven unknowns are associated with each panel: $E_x, E_y, E_z, (\partial Ex)/(\partial n), (\partial Ey)/(\partial n), (\partial Ez)/(\partial n)$, and ρ. The scalar potential ϕ is associated with the panel vertexes,

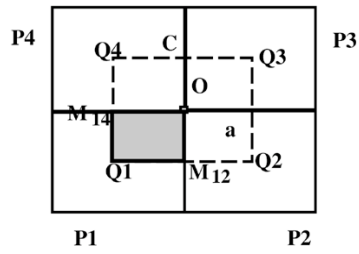


Fig. 2. Dual panel.

and $\nabla\phi$ in (30) is computed using finite-difference. With this setting, (26), (27), (28), and (29) become simple algebraic equations. But (25) deserves more attention.

Applying the integral form of (25) to the surface of an infinitely thin small rectangular box beneath the conductor surface, we obtain

$$\int_C d\gamma\vec{E}_t(\gamma)\cdot(\hat{n}(\gamma)\times\hat{l}(\gamma)) - \int_a dS(\vec{r})\hat{n}(\vec{r})\cdot\frac{\partial\vec{E}(\vec{r})}{\partial n(\vec{r})} = 0 \tag{31}$$

where a is the top of the box, C is the periphery of a, $\hat{n}$ is the normal unit vector, and $\hat{l}$ is the unit vector along C. Equation (31) is enforced on the so-called dual panel around each vertex; one dual panel $Q_1Q_2Q_3Q_4$ is shown in Fig. 2. Panel $Q_1Q_2Q_3Q_4$ is divided by the edges of regular panels into four sub-panels. The state variables $\vec{E}$ and $(\partial\vec{E})/(\partial n)$ on sub-panel $OM_{12}Q_1M_{14}$, the shaded one in Fig. 2, is the same as those defined on panel P_1. The same holds true for other three sub-panels.

Now we can write the system matrix as (32), shown at the bottom of the page, where $\phi_{\rm nc}$ and ϕ_c are the potential on the noncontacts and the contacts, respectively. The horizontal lines in the system matrix are used to mark the corresponding relation between row blocks and the equations (22)–(30). For example, the three rows in the first row block correspond to (22). Matrix I is the identity matrix, P_0 and P_1 are, respectively, the dense matrices corresponding to the single-layer integral with Green's

$$\left[\begin{array}{ccc|ccc|ccc}
P_1 & 0 & 0 & D_1 & 0 & 0 & 0 & 0 & 0\\
0 & P_1 & 0 & 0 & D_1 & 0 & 0 & 0 & 0\\
0 & 0 & P_1 & 0 & 0 & D_1 & 0 & 0 & 0\\
\hline
T_{1,x}P_0 & T_{1,y}P_0 & T_{1,z}P_0 & T_{1,x}D_0 & T_{1,y}D_0 & T_{1,z}D_0 & g_{11} & g_{12} & 0\\
T_{2,x}P_0 & T_{2,y}P_0 & T_{2,z}P_0 & T_{2,x}D_0 & T_{2,y}D_0 & T_{2,z}D_0 & g_{21} & g_{21} & 0\\
\hline
0 & 0 & 0 & 0 & 0 & 0 & -I\epsilon & 0 & P_0^1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -I\epsilon & P_0^2\\
\hline
-A_x & -A_y & -A_z & C_x & C_y & C_z & 0 & 0 & 0\\
\hline
0 & 0 & 0 & N_x & N_y & N_z & 0 & 0 & \frac{-j\omega}{\sigma}I\\
\hline
0 & 0 & 0 & T_{1,x} & T_{1,y} & T_{1,z} & 0 & 0 & 0\\
0 & 0 & 0 & T_{2,x} & T_{2,y} & T_{2,z} & 0 & 0 & 0\\
\hline
N_x & N_y & N_z & 0 & 0 & 0 & 0 & 0 & 0\\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0
\end{array}\right]
\left[\begin{array}{c}
\frac{\partial E_x}{\partial n}\\
\frac{\partial E_y}{\partial n}\\
\frac{\partial E_z}{\partial n}\\
\hline
E_x\\
E_y\\
E_z\\
\hline
\phi_{\rm nc}\\
\phi_c\\
\hline
\rho
\end{array}\right]
=
\left[\begin{array}{c}
0\\0\\0\\\hline0\\0\\0\\\hline0\\0\\\hline0\\\hline0\\\hline0\\0\\\hline0\\\hline\Phi_c
\end{array}\right] \tag{32}$$

function G_0 in (12) and G_1 in (17). The elements of matrix P_0 are

$$P_0(i,j) = \int_{\text{Panel}_j} ds(\vec{r}')G_0(\vec{r_i},\vec{r}') = \int_{\text{Panel}_j} ds(\vec{r}')\frac{e^{jk_0|\vec{r_i}-\vec{r}'|}}{4\pi|\vec{r_i}-\vec{r}'|} \tag{33}$$

where $\vec{r_i}$ is the centroid of the ith panel. Matrices P_0^1 and P_0^2 are sub-matrix blocks of P_0. The number of rows in P_0^1 and P_0^2 is the size of ϕ_{nc} and ϕ_c, respectively. D_0 and D_1 are, respectively, the dense matrices corresponding to the double-layer integral with Green's function G_0 and G_1. The elements of matrix D_0 are

$$D_0(i,j) = \begin{cases} -\int_{\text{Panel}_j} ds(\vec{r}')\frac{\partial}{\partial n(\vec{r}')}\frac{e^{jk_0|\vec{r_i}-\vec{r}'|}}{4\pi|\vec{r_i}-\vec{r}'|}, & i \neq j \\ \frac{1}{2}, & i = j. \end{cases} \tag{34}$$

Sparse matrices g_{11}, g_{12}, g_{21}, and g_{22} represent the finite-difference approximation of $\nabla\phi$. Sparse matrices $T_{1,\alpha}, T_{2,\alpha}$, and N_α $(\alpha = x, y, z)$ are the transfer matrices relating the local coordinate system (t_1, t_2, and n) to the global coordinate system (x, y, and z). The nonzero elements of the sparse matrices A_x, A_y, and A_z and the nonzero elements of the sparse matrices C_x, C_y, and C_z are related to the dual panel discretization. Φ_c, the known potential on the contact, is used as the excitation.

The structure of the system matrix in (32) can be used to show that the system is not singular even when the frequency is identically zero. It is straightforward to verify that at zero frequency the matrix block P_1 and D_1 are dense. Since the nonzero blocks P_1 in the first three columns are never in the same row, the first three columns are linearly independent. For the same reason, the fourth to sixth columns are linearly independent from each other. Noticing that the nonzero matrix blocks in rows 9 to 12 of the columns 4 to 6 are never in the same row as the nonzero blocks in rows 9 to 12 of the columns 1 to 3, we conclude that the first six columns are linearly independent. Similarly, due to the nonzero blocks $-I\epsilon$ in rows 6 and 7, we can also conclude that the first eight columns are linearly independent. At zero frequency, the matrix block $-(j\omega)/(\sigma)I$ is zero. But because of the matrix block I in column 8, columns 8 and 9 are linearly independent. Therefore, we can conclude that the system matrix is not singular even when the frequency is identically zero. This means that our surface integral formulation does not have the kind of low-frequency problem reported in [29]. Hence, we do not need to use the so-called loop/star and loop/tree basis functions to discretize the formulation.

III. Improving the Accuracy of Panel Integration

Early experiments with the above formulation suggested that there was a low-frequency problem which was resolved using a linearization technique [43]. In this section, we show that the formulation does not have difficulties at low frequencies, and that the early experimental results were due to inaccuracy in the approach to the panel integration, particularly the nearby interactions at low frequencies. We then propose a simple piecewise quadrature scheme to fix the problem.

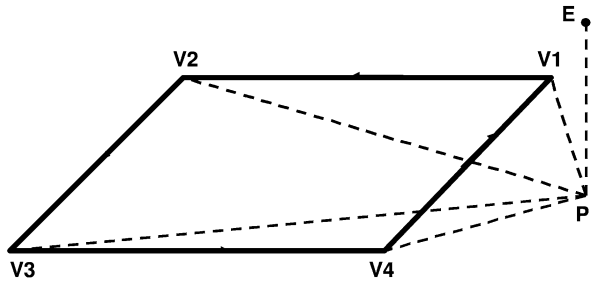


Fig. 3. Decomposition of an integration over a polygon into several integrations over triangles. Point E is the evaluation point, point P is the projection of E on the plane. $\text{Vi}\,(i = 1, 2, \ldots, 4)$ are the vertexes of the panel.

A. Definition

After discretization, the integrals over conductor surface S or S_i are replaced by the summation of integrals over panels. These integrals are

$$I_1(\vec{r}) = \int_{P_i} dS' G(\vec{r},\vec{r}') = \int_{P_i} dS' \frac{e^{jk_0|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|} \tag{35}$$

$$I_2(\vec{r}) = \int_{P_i} dS' \frac{\partial G(\vec{r},\vec{r}')}{\partial n(\vec{r}')} = \begin{cases} -\frac{1}{2} & \vec{r} \in P_i \\ \hat{n}(P_i)\cdot\int_{P_i} dS'\nabla_{\vec{r}'}G(\vec{r},\vec{r}') & \text{otherwise} \end{cases} \tag{36}$$

where P_i is the ith panel, $\hat{n}(P_i)$ is the unit normal vector on the flat panel P_i, and $G(\vec{r},\vec{r}')$ is either $G_0(\vec{r},\vec{r}')$ or $G_1(\vec{r},\vec{r}')$ defined in (12) and (17). From the symmetry property of the Green's function, it follows that

$$\int_{P_i} dS'\nabla_{\vec{r}'}G(\vec{r},\vec{r}') = -\nabla_{\vec{r}}\int_{P_i} dS' G(\vec{r},\vec{r}') = -\nabla_{\vec{r}} I_1(\vec{r}) \tag{37}$$

when $\vec{r}$ is not on panel P_i. Therefore, to compute the integrals in (35) and (36), it is only necessary to compute $I_1(\vec{r})$ and $(\partial I_1(\vec{r}))/(\partial D)$, where D stands for x, y, or z.

B. Decomposition

It is shown in [56] that any integration over a polygon is equal to the signed summation of the integration over a chosen set of triangles. The vertexes of these triangles are those of the polygon and the projection of the evaluation point onto the plane where the polygon lies, as shown in Fig. 3. To be more precise, let $f(\vec{r})$ be a general integrand, its integration over a polygon in Fig. 3 could be written as

$$\int_S d\vec{r} f(\vec{r}) = \sum_{i=1}^{N} s_i \int_{PV_iV_{i-1}} d\vec{r} f(\vec{r}) \tag{38}$$

where N is the number of vertexes, $V_{N+1} = V_1$, and $s_i = -1$ if V_iV_{i+1} is clockwise looking from the evaluation point E and $s_i = 1$ if otherwise. This idea was used in [43] to compute the integrals $I_1(\vec{r})$ and $(\partial I_1(\vec{r}))/(\partial D)$.

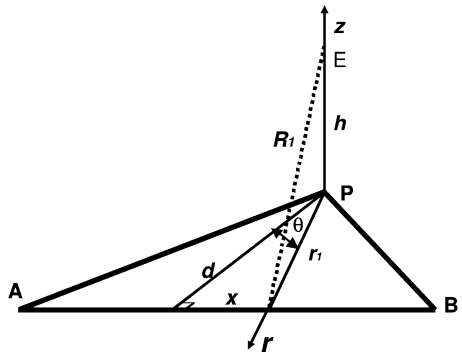


Fig. 4. Triangle in polar coordinate system, d is the distance between point P and edge AB.

C. Desingularization and Reduction to 1-D Integration

In a polar coordinate system, a triangle after the decomposition is shown in Fig. 4. Using the relation $R = \sqrt{r^2 + h^2}$ and $R\,dR = r\,dr$, the integrals I_1 and $(\partial I_1)/(\partial D)$ over this triangle could be rewritten in polar coordinates as [56]

$$\begin{aligned} I_1 &= \int_{\theta_A}^{\theta_B} d\theta \int_0^{r_1(\theta)} r\,dr \frac{e^{ikR}}{4\pi R} \\ &= \int_{\theta_A}^{\theta_B} d\theta \int_h^{R_1(\theta)} dR \frac{e^{ikR}}{4\pi} \\ &= \begin{cases} \int_{\theta_A}^{\theta_B} d\theta \frac{e^{ikR_1(\theta)} - e^{ikh}}{4\pi ik} & k \neq 0 \\ \int_{\theta_A}^{\theta_B} d\theta \frac{R_1(\theta) - h}{4\pi} & k = 0 \end{cases} \end{aligned} \tag{39}$$

$$\frac{\partial I_1}{\partial D} = \int_{\theta_A}^{\theta_B} d\theta \left(\frac{e^{ikR_1(\theta)}}{4\pi} \frac{\partial R_1(\theta)}{\partial D} - \frac{e^{ikh}}{4\pi} \frac{\partial h}{\partial D} \right). \tag{40}$$

Now the singularity of the original kernels in I_1 and $(\partial I_1)/(\partial D)$ has been eliminated and the 2-D integrations have been reduced to 1-D integrations. The quadrature rule is used to compute the two 1-D integrations in (39) and (40). The shared rapid changing kernel in these two integrals is $f(\theta) = e^{ikR_1(\theta)}$, where $R_1(\theta) = \sqrt{d^2 \sec^2(\theta) + h^2}$. When $d \ll AB, \theta_A \approx (-\pi/2)$ and $\theta_B \approx (\pi/2)$, and $f(\theta)$ changes rapidly over the interval. Many quadrature points must be used to achieve reasonable accuracy.

D. Piecewise Quadrature Scheme

A simple variable transformation and a piecewise quadrature scheme can be used to solve the above-mentioned problem. Let $x = d\tan(\theta)$, it easily follows that $(d\theta/dx) = (d/r^2)$, where $r^2 = d^2 + x^2$. The rapidly changing part of I_1 and $(\partial I_1)/(\partial D)$ could be rewritten as

$$\int_{\theta_A}^{\theta_B} d\theta e^{ikR} = \int_{x_A}^{x_B} dx\, g(x) \tag{41}$$

where $g(x) = d/(d^2 + x^2) e^{ik\sqrt{h^2 + d^2 + x^2}}$. The magnitude of the integrand $g(x)$ is shown in Fig. 5, where k is the wave number corresponding to the low-frequency $f = 1$ Hz in free space. Accurate evaluation requires many quadrature points because of the rapid variation about $x = 0$. However, dividing the integration domain into two sub-domains at $x = 0$ and using a quadrature scheme for each subdomain dramatically reduces the needed number of quadrature points. The convergence behavior of the integration over the whole domain and over the two sub-domains is shown in Fig. 6. It is clear that the piecewise scheme uses many fewer quadrature points, or has higher accuracy if only a small number of quadrature points are used. As will be shown in the numerical result section, using the piecewise scheme has indeed fixed the low-frequency problem reported in [43].

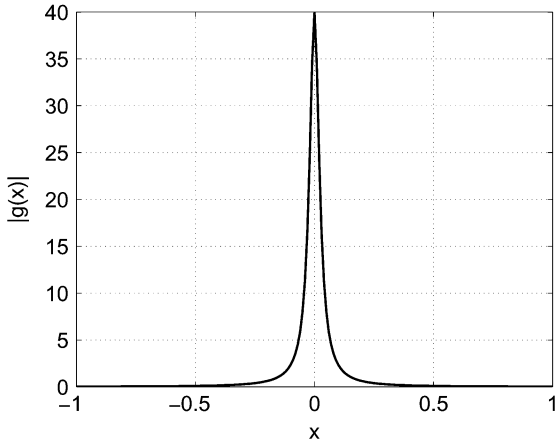


Fig. 5. Distribution of the integrand.

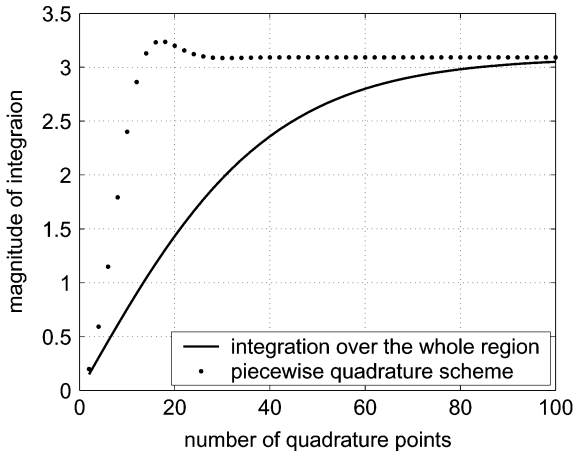


Fig. 6. Convergence behavior of different schemes.

IV. Scaling and Preconditioning

A. Scaling

The system in (32) will be solved iteratively, and therefore, reducing the system's condition number will likely accelerate iteration convergence. As expressed in (32), the system has a condition number that rises rapidly as the problem geometry decreases. The difficulty is easily eliminated by scaling.

Suppose the average edge length of panels is $O(u)$, we will first estimate the scale of each matrix block in (32) in terms of u.

The elements of matrix P_0 are expressed in (33). Since $\vec{r}_i$ and $\vec{r}'$ are on the conductor surface, it is clear that $(e^{jk_0|\vec{r}_i - \vec{r}'|})/(|\vec{r}_i - \vec{r}'|)$ in (33) is $O(1/u)$ and $ds(\vec{r}')$ is $O(u^2)$. Hence, $P_0(i,j)$ is $O(u)$. And the same holds true for the elements in matrix P_1. Following a similar reasoning, $D_0(i,j)$ in (34) is $O(1)$, as are the elements in matrix D_1.

The dual panel discretization in (31) implies that the elements in matrices C_x, C_y, and C_z are $O(u)$ and the elements in matrices A_x, A_y, and A_z are $O(u^2)$. It is easy to check that the elements in the finite difference matrices g_{11}, g_{12}, g_{21}, and g_{22} are $O(1/u)$.

Now it is clear that the scale in different matrix blocks in (32) could be different by many orders of magnitude if u is small. The huge difference in the scale could lead to large condition number. For example, the condition number could be as large as 10^{20} for micrometer feature sizes.

A simple scaling manipulation as the following can be used to remedy the situation: scale the first three columns with $1/u$ and the seventh and eighth column with u, and then scale the sixth, seventh, eighth, and the last row with $1/u$, and scale the second to the last row with u. This manipulation can be written as (42), shown at the bottom of the page, where the corresponding scale of each matrix block is also shown. It is easy to check that all matrix blocks are $O(1)$. Hence, the new system matrix is much better conditioned.

The scaling factor u could be either the average panel size or the maximum panel size. From our experiments, either one could effectively reduce the condition number. It should be pointed out that the above scaling procedure is effective only when the panels do not vary significantly in size. Otherwise, a fixed scaling factor may not be sufficient. For example, we might have to use a separate scaling factor for each column of P_1 and P_0. We shall not give more details here.

Empirical study of a simple straight wire is used here to verify the effectiveness of the scaling technique. The iterative solver GMRES [57] is used to solve the linear system matrix generated for different structure sizes. Comparison of the convergence behavior with or without the scaling is shown in Fig. 7. It is clear from this figure that the scaling has made the number of iterations almost independent of the structure size. In particular, the number of iterations has been reduced by a factor of five when the feature size is in the order of micrometers. This confirms the analysis carried out before and verifies our scaling technique.

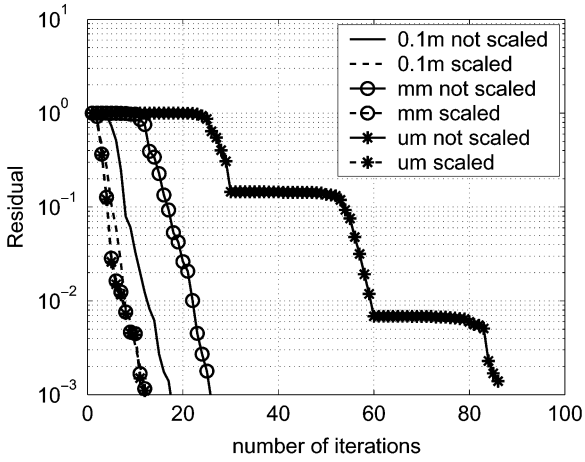


Fig. 7. Convergence behavior of the iterative solver GMRES for different structure feature sizes with or without scaling. All the dashed (solid) lines are the cases with (without) scaling. The lines with circles (stars) are for the millimeter (micrometer) sized structures, and the lines without any mark are for the structures with feature size in the order of 0.1 m.

B. Preconditioning

A straightforward way of constructing a preconditioner for the system matrix like the one in (32) is to simply replace the dense matrix blocks corresponding to the integral operators with their diagonal elements and keep all other sparse matrix blocks. This method was used in [42] to construct a preconditioner from the system matrix in (32), as shown in (43), shown at the bottom of the next page, where the superscript D means that the matrix block is just the diagonal part of the corresponding block in (32). For example, $P_0^D = \mathrm{diag}(P_0)$. Extensive numerical experiments have shown that this preconditioner significantly reduces the number of iterations. But for some structures the number of nonzeros in the preconditioner after the sparse LU factorization is still rather large. This is partially because some rows in the preconditioner before the LU factorization are not sparse enough.

$$
\left[\begin{array}{ccc|ccc|ccc}
\frac{1}{u}P_1(u) & 0 & 0 & D_1(1) & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{u}P_1(u) & 0 & 0 & D_1(1) & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{u}P_1(u) & 0 & 0 & D_1(1) & 0 & 0 & 0 \\
\hline
\frac{1}{u}T_{1,x}P_0(u) & \frac{1}{u}T_{1,y}P_0(u) & \frac{1}{u}T_{1,z}P_0(u) & T_{1,x}D_0(1) & T_{1,y}D_0(1) & T_{1,z}D_0(1) & ug_{11}\left(\frac{1}{u}\right) & ug_{12}\left(\frac{1}{u}\right) & 0 \\
\frac{1}{u}T_{2,x}P_0(u) & \frac{1}{u}T_{2,y}P_0(u) & \frac{1}{u}T_{2,z}P_0(u) & T_{2,x}D_0(1) & T_{2,y}D_0(1) & T_{2,z}D_0(1) & ug_{21}\left(\frac{1}{u}\right) & ug_{22}\left(\frac{1}{u}\right) & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & -\epsilon I & 0 & \frac{1}{u}P_0^1(u) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\epsilon I & \frac{1}{u}P_0^2(u) \\
\hline
-\frac{A_x(u^2)}{u^2} & -\frac{A_y(u^2)}{u^2} & -\frac{A_z(u^2)}{u^2} & \frac{1}{u}C_x(u) & \frac{1}{u}C_y(u) & \frac{1}{u}C_z(u) & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & N_x & N_y & N_z & 0 & 0 & \frac{-j\omega}{\sigma}I \\
\hline
0 & 0 & 0 & T_{1,x} & T_{1,y} & T_{1,z} & 0 & 0 & 0 \\
0 & 0 & 0 & T_{2,x} & T_{2,y} & T_{2,z} & 0 & 0 & 0 \\
\hline
N_x & N_y & N_z & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0
\end{array}\right] \tag{42}
$$

As explained in Section II-B, the boundary conditions, (25)–(28), are enforced in the local coordinate system. It is explained in Section II-D that (30) has to be enforced in the local coordinate system. On the other hand, the vector unknowns $\vec{E}$ and $(\partial\vec{E})/(\partial n)$ in (32) are defined in the global coordinate system. This inconsistency introduces a large number of nonzeros into (43). These nonzeros are mainly transformation between the local and the global coordinates. In addition, the diagonal elements of matrix blocks $T_{1,x}$ and $T_{2,y}$ could be zero. If only these two diagonal blocks are kept in rows 4 and 5 of the matrix in (44), shown at the bottom of the page, some of the elements in $T_{1,x}$ and $T_{2,y}$ will almost certainly be zero, and the inevitable pivoting in LU factorization will constrain the ordering algorithm used to minimize fill-ins, resulting in very dense LU factors. In order to avoid the pivoting or at least minimize it, the off-diagonal blocks in row 4 and 5 of the preconditioner in (43) have to be kept. This accounts for a large number of nonzeros.

One way to reduce the number of nonzeros is to define all vector variables and enforce all vector equations in the local coordinate system. The resulting system matrix is shown in (45), shown at the bottom of the next page, where matrices $T_{mn}(m,n=1,2,3))$ are defined as $T_{11}(i,j)=\hat{t}_1^{(i)}\cdot\hat{t}_1^{(j)}$, $T_{12}(i,j)=\hat{t}_1^{(i)}\cdot\hat{t}_2^{(j)}$, $T_{13}(i,j)=\hat{t}_1^{(i)}\cdot\hat{n}^{(j)}$, and etc., and $(\hat{t}_1^{(i)},\hat{t}_2^{(i)},\hat{n}^{(i)})$ is the local coordinate system on the ith panel. The new system matrix in (45) is different from the one in (32) by just a similarity transformation. Hence, they all have the same condition number and lead to the same convergence behavior if an iterative solver is used. But the preconditioners constructed from these different system matrices are significantly different, particularly in matrix sparsity.

TABLE I
PERFORMANCE OF PRECONDITIONERS IN THE GLOBAL AND THE LOCAL COORDINATE SYSTEM

	local	global
number of nonzeros before LU	320376	547636
number of nonzeros after LU	1097973	1318222
number of fill-in's	777597	770586
number of GMRES iterations	15	15

An important advantage of defining variables and enforcing equations in the local coordinate system is that the diagonal elements of matrix blocks T_{11}, T_{22}, and T_{33} are 1. Hence, unlike the preconditioner in (43), all the off-diagonal blocks could be thrown away. Extracting the diagonal part of the diagonal matrix blocks and keeping the remaining sparse matrices in (45) yields a new preconditioner, (46), shown at the bottom of the next page. Comparing to the preconditioner in (43), this preconditioner is much sparser. The density of most rows below the third row have been reduced by about one half.

To verify the effectiveness of the new and sparser preconditioner, we used it in the analysis of a four-turn spiral over a lossy substrate ground plane. The total number of unknowns is 72 531. The performance of the preconditioners in the global and the local coordinate system is compared in Table I. As it is expected, the preconditioner in the local coordinate system before the LU factorization is indeed much sparser. But this advantage is somewhat offset by the fill-ins generated by LU factorization. Table I

$$\left[\begin{array}{ccc|ccc|cc|c}
P_1^D & 0 & 0 & -\frac{1}{2}I & 0 & 0 & 0 & 0 & 0 \\
0 & P_1^D & 0 & 0 & -\frac{1}{2}I & 0 & 0 & 0 & 0 \\
0 & 0 & P_1^D & 0 & 0 & -\frac{1}{2}I & 0 & 0 & 0 \\
\hline
T_{1,x}P_0^D & T_{1,y}P_0^D & T_{1,z}P_0^D & \frac{1}{2}T_{1,x} & \frac{1}{2}T_{1,y} & \frac{1}{2}T_{1,z} & g_{11} & g_{12} & 0 \\
T_{2,x}P_0^D & T_{2,y}P_0^D & T_{2,z}P_0^D & \frac{1}{2}T_{2,x} & \frac{1}{2}T_{2,y} & \frac{1}{2}T_{2,z} & g_{21} & g_{22} & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & -I\epsilon & 0 & \left(P_0^1\right)^D \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -I\epsilon & \left(P_0^2\right)^D \\
\hline
-A_x & -A_y & -A_z & C_x & C_y & C_z & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & N_x & N_y & N_z & 0 & 0 & \frac{-j\omega}{\sigma}I \\
\hline
0 & 0 & 0 & T_{1,x} & T_{1,y} & T_{1,z} & 0 & 0 & 0 \\
0 & 0 & 0 & T_{2,x} & T_{2,y} & T_{2,z} & 0 & 0 & 0 \\
\hline
N_x & N_y & N_z & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0
\end{array}\right] \tag{43}$$

$$\left[\begin{array}{ccc|ccc|cc|c}
T_{1,x}P_0^D & 0 & 0 & \frac{1}{2}T_{1,x} & 0 & 0 & g_{11} & g_{12} & 0 \\
0 & T_{2,y}P_0^D & 0 & 0 & \frac{1}{2}T_{2,y} & 0 & g_{21} & g_{22} & 0
\end{array}\right] \tag{44}$$

shows that the number of nonzeros in the local preconditioner after the LU factorization is about 20% fewer than that in the global preconditioner. This directly translates into a 20% saving in memory usage. It is worth noting that though more matrix blocks are thrown away in constructing the local preconditioner than the global preconditioner, both preconditioners lead to the same iteration count, as shown in Table I.

V. Precorrected FFT Algorithm for Kernels in the Surface Formulation

After discretization, the algebraic equations (26), (27), (28), (29), and (31) become sparse matrix equations. But integral equations (22), (24), and (30) become dense matrix equations. So solving the whole system using iterative methods still takes $O(N^2)$ operations, where N is the number of unknowns. Many fast algorithms avoid forming matrix A explicitly and compute the matrix vector product approximately, requiring only $O(N)$ or $O(N\log(N))$ operations [41], [45], [58]. In this paper, we use the precorrected FFT algorithm to accelerate the dense matrix vector product corresponding to the discretized integral operators in (22), (24), and (30).

FFT-based methods are well known [59], [60], but older algorithms required a regular discretization mesh, which is not always possible or optimal for 3-D geometries. The precorrected FFT (pFFT) algorithm was originally proposed in [3], [45], and [61], where the detailed steps to accelerate a single-layer integral operator were shown. It has also been extended to the case where higher order basis functions are used [62]. The basic idea of pFFT is to separate the potential computation into far-field part and near-field part. The far-field potential is computed by using the grid charges on a uniform 3-D grid to represent charges

$$
\left[\begin{array}{ccc|ccc|cc|c}
T_{11}P_1 & T_{12}P_1 & T_{13}P_1 & T_{11}D_1 & T_{12}D_1 & T_{13}D_1 & 0 & 0 & 0 \\
T_{21}P_1 & T_{22}P_1 & T_{23}P_1 & T_{21}D_1 & T_{22}D_1 & T_{23}D_1 & 0 & 0 & 0 \\
T_{31}P_1 & T_{32}P_1 & T_{33}P_1 & T_{31}D_1 & T_{32}D_1 & T_{33}D_1 & 0 & 0 & 0 \\
\hline
T_{11}P_0 & T_{12}P_0 & T_{13}P_0 & T_{11}D_0 & T_{12}D_0 & T_{13}D_0 & g_{11} & g_{12} & 0 \\
T_{21}P_0 & T_{22}P_0 & T_{23}P_0 & T_{21}D_0 & T_{22}D_0 & T_{23}D_0 & g_{21} & g_{22} & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & -I\epsilon & 0 & P_0^1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -I\epsilon & P_0^2 \\
\hline
0 & 0 & -A_n & C_{t_1} & C_{t_2} & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & I & 0 & 0 & \frac{-j\omega}{\sigma} I \\
\hline
0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0
\end{array}\right] \tag{45}
$$

$$
\left[\begin{array}{ccc|ccc|cc|c}
P_1^D & 0 & 0 & -\frac{1}{2}I & 0 & 0 & 0 & 0 & 0 \\
0 & P_1^D & 0 & 0 & -\frac{1}{2}I & 0 & 0 & 0 & 0 \\
0 & 0 & P_1^D & 0 & 0 & -\frac{1}{2}I & 0 & 0 & 0 \\
\hline
P_0^D & 0 & 0 & \frac{1}{2}I & 0 & 0 & g_{11} & g_{12} & 0 \\
0 & P_0^D & 0 & 0 & \frac{1}{2}I & 0 & g_{21} & g_{22} & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & -I\epsilon & 0 & \left(P_0^1\right)^D \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -I\epsilon & \left(P_0^2\right)^D \\
\hline
0 & 0 & -A_n & C_{t_1} & C_{t_2} & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & I & 0 & 0 & \frac{-j\omega}{\sigma} I \\
\hline
0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0
\end{array}\right] \tag{46}
$$

on the panels. The near-field potential is computed directly. The algorithm has four steps: Projection, Convolution, Interpolation, and Nearby interaction. The effect of this algorithm is to replace the matrix vector product $A\bar{\alpha}$ with $(D + WHP)\bar{\alpha}$, where D is the direct matrix that represents the nearby interaction, W is the interpolation matrix, H is the convolution matrix, and P is the projection matrix. Matrices D, W, and P are sparse, hence their memory usage is $O(N_p)$, where N_p is the number of panels, and their product with a vector only needs $O(N_p)$ work. The matrix H is a multilevel Toeplitz matrix. Hence, its memory usage is $O(N_g)$ and its product with a vector can be computed using FFTs in $O(N_g \log(N_g))$ operations [63], where N_g is the number of grid points. Therefore, the overall computational complexity to compute $(D + WHP)\bar{\alpha}$ is $O(N_p) + O(N_g \log(N_g))$.

The surface formulation in Section II has four different kernels with two different Green's functions. All four kernels can be cast into a compact form

$$K(\vec{r}', \vec{r}) = \mathcal{F}_1(\mathcal{F}_2(G(\vec{r}', \vec{r}))) \tag{47}$$

where $G(\vec{r}', \vec{r})$ is the Green's function, and the possible options for operator $\mathcal{F}_1(\cdot)$ and $\mathcal{F}_2(\cdot)$ are

$$\mathcal{F}_1(\cdot) = U(\cdot), \frac{\partial(\cdot)}{\partial x(\vec{r})}, \frac{\partial(\cdot)}{\partial y(\vec{r})}, \frac{\partial(\cdot)}{\partial z(\vec{r})}, \frac{\partial(\cdot)}{\partial n(\vec{r})} \tag{48}$$

$$\mathcal{F}_2(\cdot) = U(\cdot), \frac{\partial(\cdot)}{\partial x(\vec{r}')}, \frac{\partial(\cdot)}{\partial y(\vec{r}')}, \frac{\partial(\cdot)}{\partial z(\vec{r}')}, \frac{\partial(\cdot)}{\partial n(\vec{r}')} \tag{49}$$

and $U(\cdot)$ is the identity operator. Treating each kernel the same way as the single-layer kernel in [3] would result in four sets of interpolation, projection, direct and convolution matrices. Notice that the two kernels in (22) and the two kernels in (30) share the same set of source/evaluation panels, one might suspect that they should share some of the four pFFT matrices as well. In this section, we show that this is indeed possible, provided the polynomial interpolation is used in both interpolation and projection step of the pFFT procedure. The algorithm presented here is general enough to apply to any integral operators in the form of (47), discretized either by a collocation or a Galerkin method using either piecewise constant or high-order basis functions.

A. Interpolation Matrix

Interpolation is the third step in the four-step pFFT algorithm, but simplest to describe. Fig. 8 shows a 2-D pictorial representation of the interpolation step, where the 3×3 uniform grid is called interpolation stencil in this paper and more points could be used to get more accurate results. The triangle inside the grid represents the local support of the testing function in Galerkin method. Suppose the potential on the uniform grids has been computed through the first two steps, namely the projection and the convolution, we could use a simple polynomial interpolation scheme to compute the potential at any point within the region covered by the grids. The potential can be expressed as [45], [48]

$$\phi(\vec{r}) = \bar{D}_0^t(\vec{r})\bar{\phi}_g \tag{50}$$

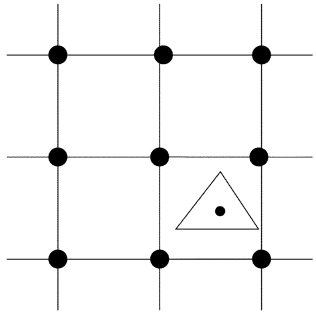


Fig. 8. 2-D pictorial representation of the interpolation step, the interpolation stencil size is 3.

where vector $\bar{D}_0^t(\vec{r})$ contains the location-dependent weighting coefficients, and superscript t denotes matrix transpose. It easily follows that the derivative of the potential at a point $\vec{r}$ with respect to α is

$$\begin{aligned} \frac{\partial \phi(\vec{r})}{\partial \alpha} &= \frac{\partial}{\partial \alpha} \bar{D}_0^t(\vec{r})\bar{\phi}_g \\ &= \bar{D}_\alpha^t(\vec{r})\bar{\phi}_g \end{aligned} \tag{51}$$

where α stands for x, y, or z. Hence, the gradient of the potential is

$$\nabla\phi(\vec{r}) = \left(\hat{x}\bar{D}_x^t(\vec{r}) + \hat{y}\bar{D}_y^t(\vec{r}) + \hat{z}\bar{D}_z^t(\vec{r})\right)\bar{\phi}_g \tag{52}$$

and the normal derivative of the potential at point $\vec{r}$ is

$$\begin{aligned} \frac{\partial \phi(\vec{r})}{\partial n} &= \hat{n} \cdot \nabla\phi(\vec{r}) = \left(n_x\bar{D}_x^t(\vec{r}) + n_y\bar{D}_y^t(\vec{r}) + n_z\bar{D}_z^t(\vec{r})\right)\bar{\phi}_g \\ &= \bar{D}_n^t(\vec{r})\bar{\phi}_g \end{aligned} \tag{53}$$

where n_x, n_y, and n_z are the projection of the unit normal vector along x, y, and z direction. Using the notation in (48), (50), (51), and (53) could be written as

$$\mathcal{F}_1(\phi(\vec{r})) = \bar{D}_\beta^t(\vec{r})\bar{\phi}_g \tag{54}$$

where $\bar{D}_\beta^t(\vec{r})$ stands for $\bar{D}_0^t(\vec{r})$, $\bar{D}_\alpha^t(\vec{r})$, or $\bar{D}_n^t(\vec{r})$.

In Galerkin method, we want to compute

$$\Psi_i = \int_{\Delta_i^t} dS\mathcal{F}_1(\phi(\vec{r}))t_i(\vec{r}), \quad i = 1, 2, \ldots, N_t. \tag{55}$$

where Δ_i^t is the support of the testing function $t_i(\vec{r})$ and N_t is the number of testing functions. Substituting (54) into (55) yields

$$\Psi_i = \int_{\Delta_i^t} dSt_i(\vec{r})\bar{D}_\beta^t(\vec{r})\bar{\phi}_g = \left(\bar{W}_\beta^{(i)}\right)^t \bar{\phi}_g, \quad i = 1, 2, \ldots, N_t \tag{56}$$

where

$$\left(\bar{W}_\beta^{(i)}\right)^t = \int_{\Delta_i^t} dSt_i(\vec{r})\bar{D}_\beta^t(\vec{r}). \tag{57}$$

If the collocation method is used, then (57) can be simplified as

$$\left(\bar{W}_\beta^{(i)}\right)^t = \bar{D}_\beta^t\left(\vec{r}_c^{(i)}\right), \quad i = 1, 2, \ldots, N_t \tag{58}$$

where $\vec{r}_c^{(i)}$ is the ith collocation point.

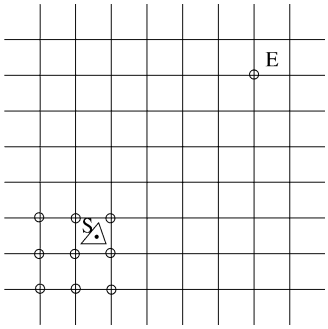


Fig. 9. 2-D pictorial representation of the projection step, the projection stencil size is 3.

In matrix format, (56) becomes

$$\bar{\Psi} = [W]\bar{\phi}_g \tag{59}$$

where $[W]$ is an $N_t \times N_g$ matrix, and N_g is the number of grid points. To cover the local support of a basis function, only a small number of interpolation grid points are needed, as shown in Fig. 8. Therefore, computing each Ψ_i through interpolation only involves grid potentials at just a few grid points. This implies that each row of the interpolation matrix $[W]$ is sparse. The nonzero elements in the ith row of the matrix $[W]$ are just the elements of the row vector $(\bar{W}_\beta^{(i)})^t$ in (57) or (58). And the number of nonzeros in each row is equal to the number of grid points in the interpolation stencil.

B. Projection Matrix

Fig. 9 shows a 2-D pictorial representation of the projection step, where the 3×3 uniform grid is called projection stencil. More points could be used if higher accuracy is desired. Similar to the previous section, a triangle is used to represent the support of a basis function.

We start with a point charge ρ_p at point $\mathbf{S}$ on the triangle, shown in Fig. 9. The potential at point $\mathbf{E}$ due to this point charge is

$$\phi_E = \rho_p G(\vec{r}_s, \vec{r}_E). \tag{60}$$

The purpose of the projection is to find a set of grid charges $\bar{\rho}_g$ on the projection grid points such that they generate approximately the same potential at point $\mathbf{E}$, i.e.,

$$\phi_E^A = \sum_i \rho_{g,i} G(\vec{r}_i, \vec{r}_E) = (\bar{\rho}_g)^t \bar{\phi}_g \simeq \phi_E \tag{61}$$

where $\phi_{g,i} = G(\vec{r}_i, \vec{r}_E)$. Due to the duality between interpolation and projection [45], the same polynomial interpolation scheme used for interpolation can also be used for projection and the projection grid charges for charge ρ_p at $\vec{r}_s$ can be expressed as [48]

$$\bar{\rho}_g = \rho_p \bar{p}(\vec{r}_s) \tag{62}$$

where column vector $\bar{p}(\vec{r}_s)$ contains the location-dependent projection coefficients.

If the kernel has a differential operator inside the integral, the potential at point $\mathbf{E}$ due to point charge ρ_p is

$$\phi_E = \rho_p \frac{\partial}{\partial \beta(\vec{r}_s)} G(\vec{r}_s, \vec{r}_E). \tag{63}$$

where β stands for x, y, z, or n. We again want to find a set of *grid monopole charges* $\bar{\sigma}_\beta$ (not the dipole charges) on the projection grid points such that they generate the same potential at point $\mathbf{E}$, i.e.,

$$\phi_E^A = \sum_i \sigma_{\beta,i} G(\vec{r}_i, \vec{r}_E) = (\bar{\sigma}_\beta)^t \bar{\phi}_g \simeq \phi_E. \tag{64}$$

Equations (60)–(62) imply that

$$G(\vec{r}_s, \vec{r}_E) \simeq \bar{p}^t(\vec{r}_s)\bar{\phi}_g. \tag{65}$$

Substituting (65) into (63) and then comparing (63) with (64) yields

$$\bar{\sigma}_\beta = \rho_p \frac{\partial \bar{p}(\vec{r}_s)}{\partial \beta(\vec{r}_s)}. \tag{66}$$

In particular, if the kernel is a double-layer kernel, then (66) becomes

$$\begin{aligned} \bar{\sigma}_n &= \rho_p \frac{\partial \bar{p}(\vec{r}_s)}{\partial n(\vec{r}_s)} \\ &= \rho_p \left(n_x \frac{\partial \bar{p}(\vec{r}_s)}{\partial x} + n_y \frac{\partial \bar{p}(\vec{r}_s)}{\partial y} + n_z \frac{\partial \bar{p}(\vec{r}_s)}{\partial z} \right) \end{aligned} \tag{67}$$

where n_x, n_y, and n_z are the projection of the unit normal vector at $\vec{r}_s$ along x, y, and z direction.

A charge distribution $b_j(\vec{r})$ on the jth basis function support could be regarded as a linear combination of an infinite number of point charges. Both (62) and (66) imply that the projection monopole charges are linearly proportional to the source point charge, monopole in (60) or dipole in (63). Hence, it easily follows that the projection charges for a monopole and a dipole charge distribution $b_j(\vec{r})$ is

$$\bar{\rho}_g^{(j)} = \int_{\Delta_j^b} dS b_j(\vec{r}) \bar{p}(\vec{r}) \tag{68}$$

$$\bar{\sigma}_\beta^{(j)} = \int_{\Delta_j^b} dS b_j(\vec{r}) \frac{\partial \bar{p}(\vec{r}_s)}{\partial \beta(\vec{r})} \tag{69}$$

respectively.

We usually have to use more than one basis function in the approximate solution. In this case, the total charge on each grid point is the accumulation of the grid charge due to each basis function. Assuming there are N_b basis functions and N_g grid points, the relation between the total grid charges $\bar{Q}_g$ and the magnitude of basis functions $\bar{\alpha}$ is

$$\bar{Q}_g = [P]\bar{\alpha}. \tag{70}$$

where $[P]$ is an $N_g \times N_b$ matrix. Due to the locality of the basis support, the projection grid for each basis function has only a small number of points. In addition, the Green's function in (65) is a smooth function of $\vec{r}$ when $|\vec{r} - \vec{r}_E|$ is large. Hence,

TABLE II
RELATION BETWEEN OPERATOR PAIR AND THE INTERPOLATION MATRIX AND THE PROJECTION MATRIX

F_1	$U(\cdot)$	$\frac{\partial(\cdot)}{\partial x'}$, $\frac{\partial(\cdot)}{\partial y'}$, $\frac{\partial(\cdot)}{\partial z'}$	$\frac{\partial(\cdot)}{\partial n'}$
interpolation	$\bar{W}_0^{(i)}$ in (57)	$\bar{W}_x^{(i)}$, $\bar{W}_y^{(i)}$, $\bar{W}_z^{(i)}$ in (57)	$\bar{W}_n^{(i)}$ in (57)
F_2	$U(\cdot)$	$\frac{\partial(\cdot)}{\partial x}$, $\frac{\partial(\cdot)}{\partial y}$, $\frac{\partial(\cdot)}{\partial z}$	$\frac{\partial(\cdot)}{\partial n}$
projection	$\bar{\rho}_g^{(j)}$ in (68)	$\bar{\sigma}_x^{(j)}$, $\bar{\sigma}_y^{(j)}$, $\bar{\sigma}_z^{(j)}$ in (69)	$\bar{\sigma}_n^{(j)}$ in (69)

low-order polynomials are sufficient to approximate it well. This implies that each column of the projection matrix $[P]$ is rather sparse. The nonzero elements in the jth column of matrix $[P]$ are the elements of the column vector $\bar{\rho}_g^{(j)}$ in (68) or $\bar{\sigma}_\beta^{(j)}$ in (69). And the number of nonzeros in each column is equal to the number of grid points in the projection stencil.

C. Summary

Sections V-A and V-B are summarized in Table II. It is clear by now that the interpolation matrix $[W]$ is only related to the operator $\mathcal{F}_1$ and the testing functions and the projection matrix $[P]$ is only related to the operator $\mathcal{F}_2$ and the basis functions. They are all independent of the Green's function.

The direct matrix, however, is dependent upon all the above information. So we have to set up one direct matrix for each $\mathcal{F}_1$ and $\mathcal{F}_2$ operator pair. The convolution matrix, on the other hand, is only related to the Green's function and the location of grid points. It is not related to $\mathcal{F}_1$ or $\mathcal{F}_2$. So we only need to set up one convolution matrix for each unique Green's function.

In light of the above observations, we only need one interpolation matrix, two projection matrices, two direct matrices and one convolution for the set of two kernels in (22) and (30), respectively.

VI. NUMERICAL RESULTS

Based upon the algorithms described in previous sections, we have developed FastImp, a fast impedance extraction program. In this section, we first use small examples to demonstrate FastImp's accuracy. We then use a few practical examples to demonstrate FastImp's flexibility. Finally, we use large examples to show FastImp's speed. If not specified explicitly, the calculations were carried out on a desktop computer with a Pentium IV microprocessor and 1-GB memory.

A. FastImp's Accuracy

Ring Example: This ring example is used to verify the piecewise quadrature scheme proposed in Section III. We also intend to use this relatively small example to conduct the convergence test of FastImp. The third goal is to numerically show that with the help of a preconditioner the formulation behaves reasonably well across a wide frequency range.

The ring is 10 mm in radius, with a square cross section of the size 0.5 mm $\times$ 0.5 mm. The conductivity is that of the copper, which is $5.8e7$ sm^{-1}. In order to compare with results from the well-established FastHenry program [2], we have carried out the magneto-quasi-static (MQS) analysis. The number of filaments used by FastHenry is 960, 3840, and 15360, respectively. And the number of panels used by FastImp is 992 and 2048,

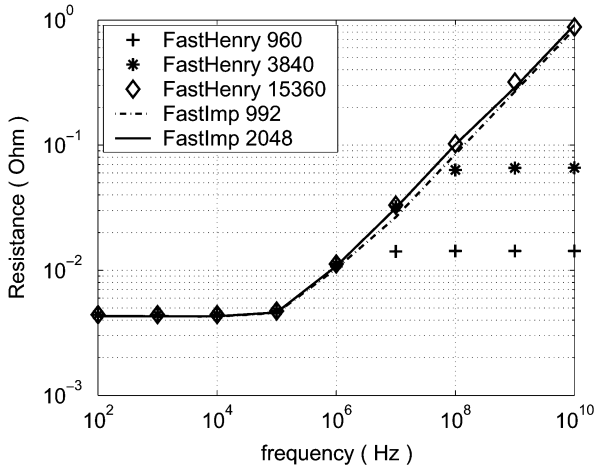


Fig. 10. Resistance of a ring.

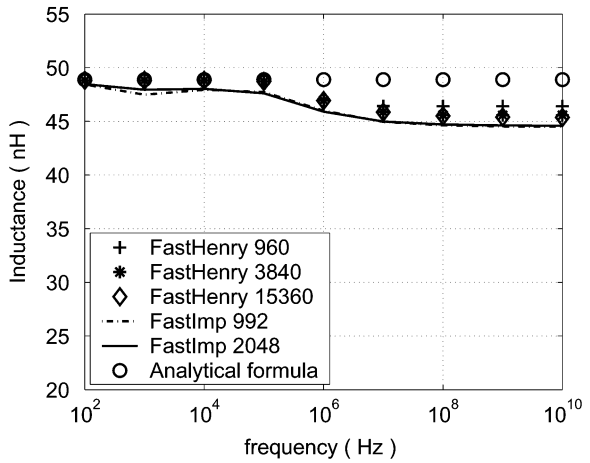


Fig. 11. Inductance of a ring.

respectively. The resistance and inductance calculated by both codes are compared in Figs. 10 and 11, where the low frequency inductance calculated using the analytical formula in [64] is 48.89 nH. These two figures show that FastImp's results converges very well with the refinement of panel discretization. It should be noted that the inductance calculated with FastImp is very close to 48.89 nH in the low frequency range, as shown in Fig. 11. This suggests that the piecewise quadrature scheme proposed in Section III has indeed eliminated the low frequency problem reported in [43]. Also, at high frequencies, the resistance in Fig. 10 scales to the square root of frequency and the inductance in Fig. 11 drops a little. This suggests that the skin effect has been well captured.

It is worth mentioning that a large number of filaments has to be used by FastHenry in order to capture the skin effect at high frequencies. On the other hand, with a small and fixed number of panels, the skin effect has been well captured by FastImp. This clearly demonstrates the advantage of the surface integral formulation over the volume integral formulation.

Fig. 12 shows the number of GMRES iterations versus frequency for discretization with 992 and 2048 panels. Here the tolerance for GMRES is set to be $1e-4$. Though low-frequency calculation takes more GMRES iterations, the number of iterations is still very reasonable, considering the number of

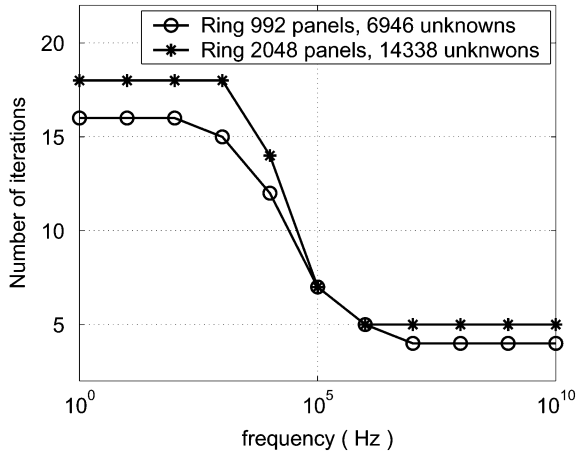


Fig. 12. Number of GMRES iterations versus frequency.

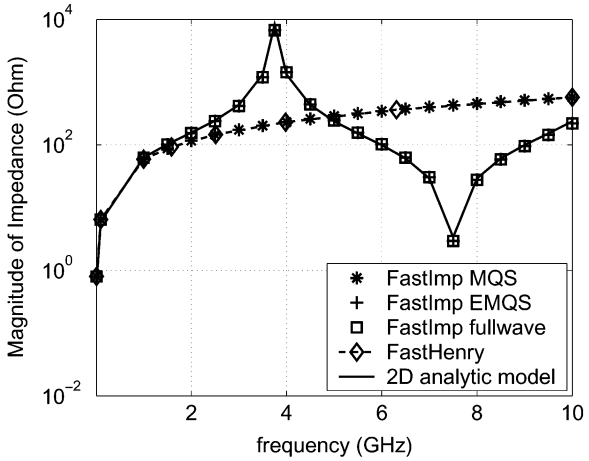


Fig. 13. Magnitude of the impedance of a shorted transmission line, length is 2 cm, separation is 50 μm.

unknowns is 6946 and 14338, respectively. This indicates that FastImp is indeed a wide-band solver.

Shorted Transmission Line: The length of the transmission line is 2 cm. The cross section of each conductor is 50×50 μm, and the space between two conductors is 50 μm. The conductivity of both conductors is again $5.8e7$ sm^{-1}. One end of this transmission line is shorted, and the other end becomes a port.

In theory, this finite-length transmission line is a 3-D structure. We have used FastImp to carry out 3-D MQS analysis, 3-D electromagneto-quasi-static (EMQS) analysis and 3-D full-wave analysis and calculated the input impedance at the open port. This finite-length transmission line could also be treated as a quasi-2-D structure since its length is 400 times its width and thickness. In this case, the per-unit-length resistance R_0 and inductance L_0 are obtained by dividing the total resistance and inductance (from the 3-D MQS analysis) by the length. And the per-unit-length capacitance C_0 is just

$$C_0 = \frac{1}{c^2 L_0} \tag{71}$$

where c is the speed of light. The behavior of a shorted 2-D transmission line is well understood. We could calculate its input impedance from R_0, C_0 and L_0 [50]. This 2-D analytic model is used as standard for EMQS results.

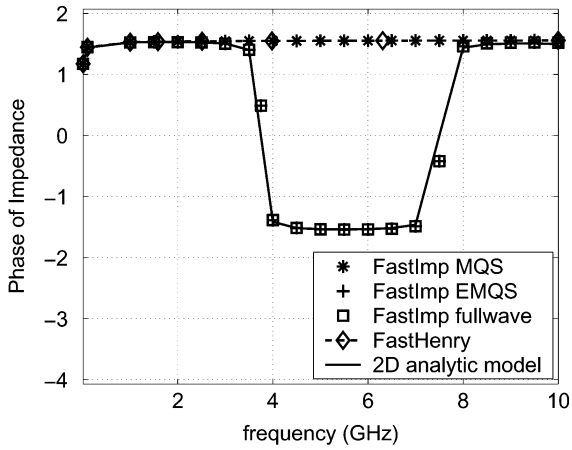


Fig. 14. Phase of the impedance of a shorted transmission line, length is 2 cm, separation is 50 μm.

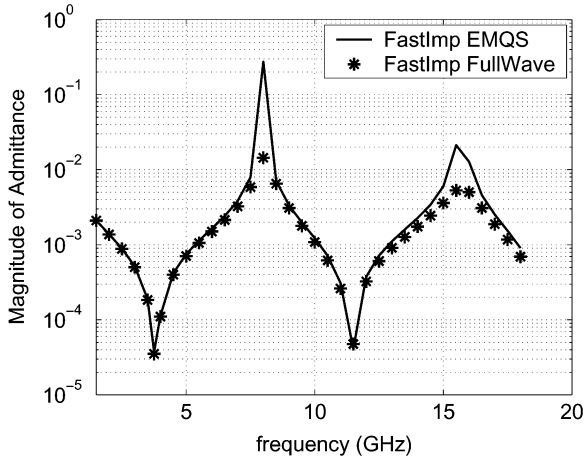


Fig. 15. Magnitude of the admittance of a shorted transmission line, length is 2 cm, separation is 1 cm.

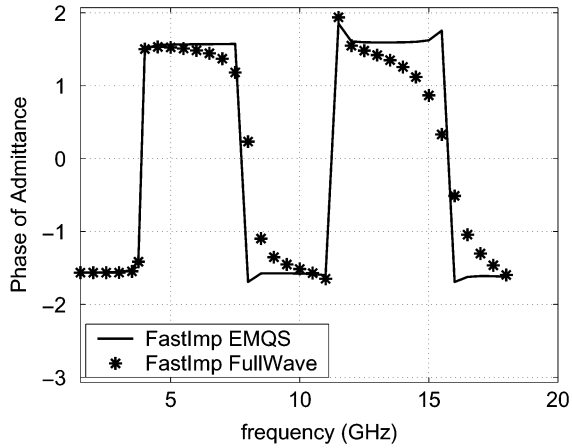


Fig. 16. Phase of the admittance of a shorted transmission line, length is 2 cm, separation is 1 cm.

We have also used FastHenry to calculate the input impedance of this shorted transmission line. We used 20×20 filaments on the cross section of each conductor. Since this rather fine discretization guarantees that the size of the filaments close to the conductor surface is smaller than the smallest skin depth in the

TABLE III
COMPARISON OF CPU TIME AND MEMORY USAGE FOR VARIOUS PRACTICAL STRUCTURES

	bus	circular spirals	rectangle spirals
#panels	18,540	15,194	18,520
#unknowns	148,380	121,558	148,166
FastImp	9min,340Mb	68min,642Mb	54min,749Mb
*iterative	160min,19GB	750min,72GB	590min,83GB
*standard	136days,19GB	100days,19GB	168days,22GB

* obtained by estimation or extrapolation.

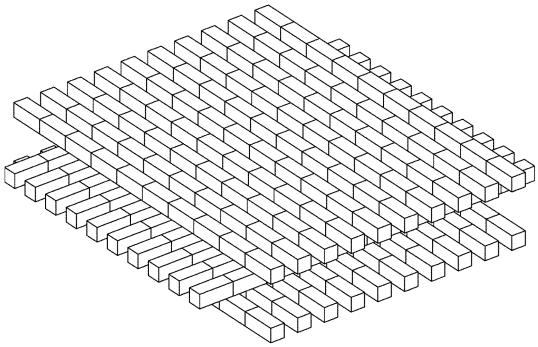


Fig. 17. Multiple conductor bus.

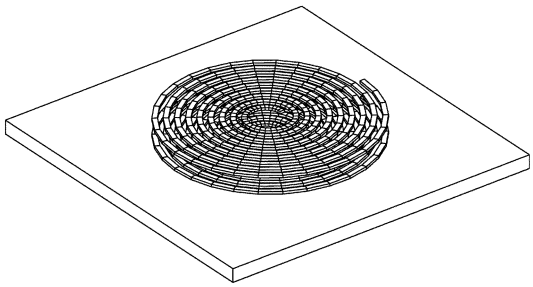


Fig. 18. Stacked nine-turn circular spirals over ground.

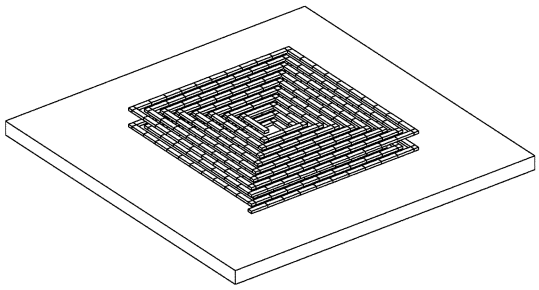


Fig. 19. Stacked eight-turn rectangular spirals over ground.

frequency range we care about, FastHenry's results should be accurate.

In order to verify FastImp's MQS, EMQS and full-wave analysis, we have plotted Figs. 13 and 14, the magnitude and phase of the input impedance calculated using different methods. FastImp's MQS results are almost identical to FastHenry's results. FastImp's EMQS results are almost indistinguishable from those of the 2-D analytic model. As expected, the EMQS

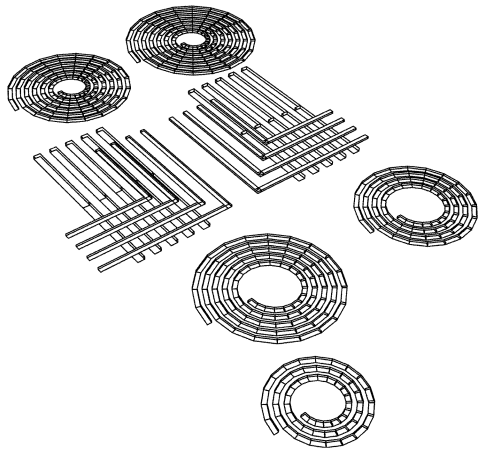


Fig. 20. Portion of an RF circuit consisting of five circular spirals and two pieces of 3-D interconnects with straight wires and right-angle bends.

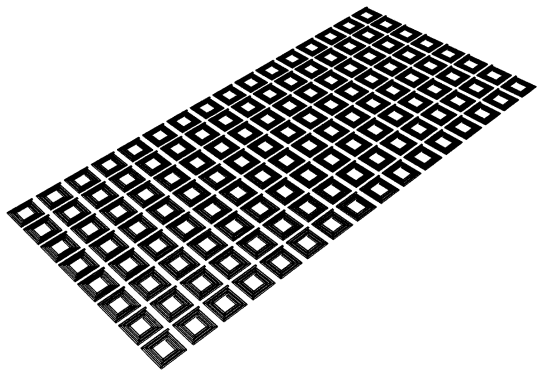


Fig. 21. 16×8 three-turn rectangular spiral array.

results are essentially the same as MQS results at lower frequencies (less than 1 GHz), and resonances appear at higher frequencies. Since the separation distance between two conductors of the transmission line is only 50 μm, a small fraction of the shortest wave length at 10 GHz, the full-wave results are essentially same as the EMQS results.

To see the difference in EMQS and full-wave results, we deliberately used a larger separation between the two conductors of the shorted transmission line. Figs. 15 and 16 show the the magnitude and phase of the input admittance calculated using EMQS and full-wave modes. Here the separation is 1 cm. Due to full-wave radiation loss at resonant frequencies, the magnitude of full-wave admittance is smaller than that of EMQS admittance at resonant frequencies, as shown in Fig. 15. For the same reason, the imaginary part of the full-wave admittance is less dominant than that of EMQS admittance. Hence, the phase change at resonant frequencies for full-wave admittance is not as sharp as that of EMQS admittance, as shown in Fig. 16. Figs. 15 and 16 are in good agreement with the experiments carried out in [52].

The comparison in Figs. 13, 14, 15, and 16 clearly demonstrates the accuracy of FastImp's MQS, EMQS and full-wave analysis.

TABLE IV
DISCRETIZATION OF THE RF INTERCONNECT EXAMPLE AND THE 16×8 SPIRAL ARRAY EXAMPLE

	RF interconnect	16x8 spiral array
number of panels	15,566	180,224
number of unknowns	124,574	1,442,048
number of grids	$32 \times 64 \times 16$	$256 \times 128 \times 8$
grid step	3.53um	1.91um

TABLE V
A DETAILED BREAKDOWN OF THE CPU TIME USED BY THE RF INTERCONNECT EXAMPLE AND THE 16×8 SPIRAL ARRAY EXAMPLE. UNIT IS SECOND

	RF interconnect	16x8 array
P and I matrices	23.5	746
D and H matrices	1444	14353
form the preconditioner P_r	3.61	53
LU factorization of P_r	3.28	1927
GMRES (tol = 1e-3)	369 (48 iter)	25168 (80 iter)
total	1820	42247

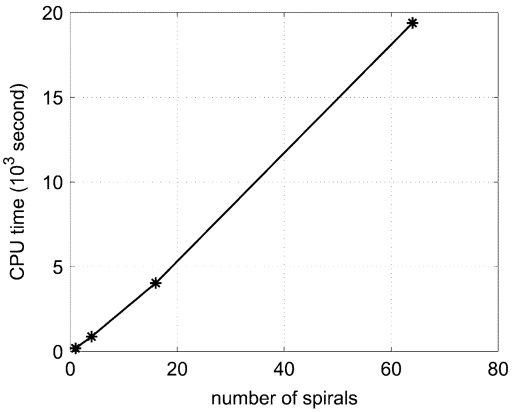


Fig. 22. CPU time versus number of spirals in the spiral arrays.

B. FastImp's Flexibility

A few practical structures are analyzed in this section to demonstrate FastImp's flexibility. The CPU time and memory usage for different examples are compared in Table III.

Multiple Conductor Crossover Bus: Fig. 17 shows a multiple conductor bus with three-layer of identical conductors. Each layer has 10 conductors and the conductors on different layer are orthogonal to each other. The size of every conductor is $1 \times 1 \times 25\ \mu$m. We only extracted one column of the impedance matrix (since this is a multiple port structure) at one frequency point $f = 1$ GHz using the EMQS analysis. The CPU time and memory usage are shown in Table III.

Stacked Spirals Over Ground: The impedance matrix of two stacked nine-turn circular spirals over a lossy ground plane (shown in Fig. 18) and two stacked eight-turn rectangular spirals over a lossy ground plane (shown in Fig. 19) are extracted at one frequency point $f = 1$ GHz using the EMQS analysis. The CPU time and memory usage are shown in Table III.

C. FastImp's Speed

Large 3-D Structures: FastImp has been used to perform the EMQS analysis of two large structures shown in Figs. 20 and 21. Fig. 20 shows a portion of an RF circuit, which includes five circular spiral inductors of various sizes and number of turns, and two 3-D interconnect structures with straight wires and right-angle bends. Fig. 21 shows a 16×8 array of three-turn rectangular spirals. The discretization, detailed breakdown of CPU time and memory usage for the analysis of these two examples are shown in Tables IV, V, and VI, respectively. The analysis of the 16×8 spiral array in Fig. 21 was carried out on a server with 32 GB memory and one 64-bit Itanium microprocessor. This server is about 3 times slower than the desktop computer used for other examples.

TABLE VI
A DETAILED BREAKDOWN OF THE MEMORY USAGE FOR THE RF INTERCONNECT EXAMPLE AND THE 16×8 SPIRAL ARRAY EXAMPLE. UNIT IS GB

	RF interconnect	16x8 spiral array
direct matrices	0.593	5.54
projection matrices	0.025	0.39
interpolation matrices	0.015	0.23
convolution matrices	0.013	0.13
maps between grids and panels	0.026	0.70
preconditioner	0.114	2.76
GMRES	0.100	2.21
total	0.886	11.96

Computational Complexity of FastImp: We have used FastImp to analyze a series of similar structures with increasingly larger size. These structures are 1×1, 2×2, 4×4, and 8×8 spiral arrays. All elements in these arrays are three-turn rectangular spirals. The CPU time versus number of spiral elements in the spiral arrays is shown in Fig. 22. The plot clearly indicates that the CPU time grows nearly linearly with the problem size.

VII. CONCLUSION

We have derived a recently developed surface integral formulation from a different perspective. Using a piecewise quadrature scheme to improve the accuracy of panel integration, we have fixed the low-frequency problem in the original formulation. Using a scaling and a local preconditioning technique, we have improved the accuracy and memory efficiency of the formulation. We have also generalized the precorrected FFT algorithm to allow the acceleration of complicated integral operators. Based on this generalization we have developed a flexible and extensible fast integral equation solver, pfft++. With 4 to 5 digit accuracy at modest computational cost and nearly $O(N)$ computational complexity, pfft++ could be easily applied to a wide range of engineering problems. Using pfft++ as the engine, we have developed a fast impedance extraction program, FastImp. Numerical examples show that FastImp can efficiently and robustly perform wide-band electromagnetic analysis of general 3-D structures. Both pfft++ and FastImp are now available at www.rle.mit.edu/cpg/research_codes.htm.

ACKNOWLEDGMENT

The first author wishes to thank Dr. M. Tsuk at Hewlett-Packard Company. Part of this work was done when the first

author worked as a summer intern in the Alpha Server Platform Group of the Hewlett-Packard Company.

REFERENCES

[1] K. Nabors and J. White, "FASTCAP: A multipole-accelerated 3-D capacitance extraction program," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 10, no. 11, pp. 1447–1459, Nov. 1991.

[2] M. Kamon, M. J. Tsuk, and J. White, "FastHenry: A multipole-accelerated 3-D inductance extraction program," *IEEE Trans. Microwave Theory Tech.*, vol. 42, no. 9, pp. 1750–1758, Sep. 1994.

[3] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 16, no. 10, pp. 1059–1072, Oct. 1997.

[4] S. Kapur and D. Long, "IES3: A fast integral equation solver for efficient 3-dimensional extraction," in *Proc. IEEE/ACM Int. Conf. Computer Aided-Design*, 1997, pp. 448–455.

[5] M. Bachtold, M. Spasojevic, C. Lage, and P. Ljung, "A system for full-chip and critical net parasitic extraction for ULSI interconnects using a fast 3-D field solver," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 19, no. 3, pp. 325–338, Mar. 2000.

[6] J. R. Phillips, E. Chiprout, and D. D. Ling, "Efficient full-wave electromagnetic analysis via model-order reduction of fast integral transformations," in *Proc. 33rd Design Automation Conf.*, 1996, pp. 377–382.

[7] J. Song, C. Liu, W. Chew, and S. Lee, "Fast Illinois solver code (FISC)," *IEEE Antennas Propag. Mag.*, vol. 40, no. 3, pp. 27–34, Jun. 1998.

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

[9] J. Wang, *Generalized Moment Methods in Electromagnetics*. New York: Wiley, 1991.

[10] S. Rao and D. Wilton, "E-field, H-field, and combined field solution for arbitrarily shaped three dimensional dielectric bodies," *Electromagnetics*, no. 10, pp. 407–421, 1990.

[11] R. Harrington, "Boundary integral formulation for homogeneous material bodies," *J. Electromagn. Waves Applicat.*, vol. 3, no. 1, pp. 1–15, 1989.

[12] A. Krawczyk, *Numerial Modeling of Eddy Currents*. Oxford, U.K.: Clarendon, 1993.

[13] W. M. Rucker, R. Hoschek, and K. R. Richter, "Various BEM formulations for calculating eddy current in terms of field variables," *IEEE Trans. Magn.*, vol. 31, no. 3, pp. 1336–1341, May 1995.

[14] W. Chew, *Waves and Fields in Inhomogeneous Media*. New York: IEEE Press, 1995.

[15] J. Stratton, *Electromagnetic Theory*. New York: McGraw-Hill, 1941.

[16] B. Kolundzija, "Electromagnetic modeling of composite metallic and dielectric structures," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 7, pp. 566–569, Jul. 1999.

[17] T. Wu and L. Tsai, "Scattering from arbitrary-shaped lossy dielectric bodies of revolution," *Radio Sci.*, vol. 12, no. 5, pp. 709–718, 1977.

[18] L. Medgyesi-Mitschange, J. Putname, and M. Gendre, "Generalized method of moments for three-dimensional penetrable scatters," *Opt. Soc. Amer.*, vol. 11, no. 4, pp. 1383–1398, Apr. 1984.

[19] J. R. Mosig, "Arbitrarily shaped microstrip structures and their analysis with mixed potential integral equation," *IEEE Trans. Microwave Theory Tech.*, vol. 36, no. 2, pp. 314–323, Feb. 1988.

[20] D. C. Chang and J. X. Zheng, "Electromagnetic modeling of passive circuit elements in MMIC," *IEEE Trans. Microwave Theory Tech.*, vol. 40, no. 9, pp. 1741–1747, Sep. 1992.

[21] R. Bunger and F. Arndt, "Efficient MPIE approach for the analysis of three-dimensional microstrip structures in layered media," *IEEE Trans. Microwave Theory Tech.*, vol. 45, no. 8, pp. 1141–1753, Aug. 1997.

[22] F. Ling, D. Jiao, and J.-M. Jin, "Efficient electromagnetic modeling of microstrip structures in multilayer media," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 9, pp. 1810–1818, Sep. 1999.

[23] T. Nakata, N. Takahashi, K. Fujiwara, K. Muramatsa, and Z. Cheng, "Comparison of various methods for 3-D eddy current analysis," *IEEE Trans. Magn.*, vol. 24, no. 6, pp. 3159–3161, Nov. 1988.

[24] Y. M. Massoud, "Simulation algorithms for inductive effects," Ph.D. dissertation, Electr. Eng. Comput. Sci. Dept., Mass. Inst. Technol., Cambridge, MA, 1999.

[25] S. Kalaicheluan and J. Lavers, *BEM for Eddy Current Problems*, C. A. Brebbia, Ed. Berlin, Germany: Springer Verlag, 1989, vol. 6, Topics in BE Research, pp. 79–116.

[26] D. Colton and R. Kress, *Integral Equation Methods in Scattering Theory*. Malabar, FL: Krieger, 1992.

[27] A. Peterson, "The interior resonance problem associated with surface integral equations of electromagnetics: Numerical consequences and a survey of remedies," *Electromagnetics*, pp. 293–312, Oct. 1990.

[28] S. Rao, D. Wilton, and A. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, no. 3, pp. 409–418, May 1982.

[29] J. Mautz and R. Harrington, "An E-field solution for a conducting surface small or comparable to the wavelength," *IEEE Trans. Antennas Propag.*, vol. 32, no. 4, pp. 330–339, Apr. 1984.

[30] W.-L. Wu, A. W. Glisson, and D. Kajfez, "A study of two numerial solution precedures for the electric field integral equation at low frequency," *Appl. Computat. Electromagn.*, vol. 10, pp. 69–80, Nov. 1995.

[31] Y. Chu and W. Chew, "A surface integral equation method for solving complicated electrically small structures," in *Proc. IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, Nov. 2003, pp. 341–344.

[32] H. Heeb and A. E. Ruehli, "Three-dimensional interconnect analysis using partial element equivalent circuits," *IEEE Trans. Circuits Syst. I: Fund. Theory Applicat.*, vol. 39, no. 11, pp. 974–982, Nov. 1992.

[33] M. Kamon, N. Marques, L. Silveria, and J. White, "Automatic generation of accurate circuit models of 3-D interconnect," *IEEE Trans. Compon., Packag., Manufact. Technol.—Part B*, vol. 21, no. 3, pp. 225–240, Aug. 1998.

[34] M. J. Tusk and J. A. Kong, "A hybrid method for the calculation of the resistance and inductance of transmission lines with arbitrary cross sections," *IEEE Trans. Microwave Theory Tech.*, vol. 39, no. 8, pp. 1338–1347, Aug. 1991.

[35] L. Daniel, A. Sangiovanni-Vincentelli, and J. White, "Using conduction modes basis functions for efficient electromagnetic analysis of on-chip and off-chip interconnect," in *Proc. Design Automation Conf.*, Jun. 2001, pp. 563–566.

[36] E. Tuncer, B. Lee, and D. Neikirk, "Interconnect series impedance determination using a surface ribbon method," in *Proc. IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, Nov. 1994, pp. 249–252.

[37] K. M. Coperich, A. C. Cangellaris, and A. E. Ruehli, "Enhanced skin effect for partial-element equivalent-circuit (PEEC) model," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1435–1442, Sep. 2000.

[38] C. Y. Yang, G. Ouyang, and V. Jandhyala, "Integral equation based time domain coupled EM-circuit simulation for packaged conductors and dielectrics," in *Proc. IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, Nov. 2003, pp. 371–374.

[39] A. Rong, A. Cangellaris, and L. Dong, "Comprehensive broad-band electromagnetic modeling of on chip interconnects with a surface discretization-based generalized PEEC model," in *Proc. IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, Nov. 2003, pp. 367–370.

[40] L. Greengard and V. Rohklin, "A fast algorithm for particle simulations," *J. Computat. Phys.*, vol. 73, no. 2, pp. 325–348, Dec. 1987.

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

[42] J. Wang and J. K. White, "A wide frequency range surface integral formualtion for 3-D *RLC* extraction," in *Proc. Int. Conf. Computer Aided-Design*, Nov. 1999, p. 453.

[43] J. Wang, "A new surface integral formulation of EMQS impedance extraction for 3-D structures," Ph.D. dissertation, Electr. Eng. Comput. Sci. Dept., Mass. Inst. Technol., Cambridge, MA, 1999.

[44] Z. Zhu, J. Huang, B. Song, and J. K. White, "Improving the robustness of a surface integral formulation for wide-band impendance extraction of 3-D structures," in *Proc. Int. Conf. Computer Aided-Design*, Nov. 2001, pp. 592–597.

[45] J. R. Phillips, "Rapid solution of potential integral equations in complicated 3-Dimensional geometries," Ph.D. dissertation, Electr. Eng. Comput. Sci. Dept., Mass. Inst. Technol., Cambridge, MA, 1997.

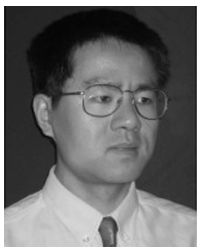
[46] X. Wang, P. Mucha, and J. White, "Fast fluid analysis for multibody micromachined devices," in *Proc. Int. Conf. Modeling and Simulation of Microsystems*, Hilton Head Island, SC, 2001, pp. 19–22.

[47] S. Kuo, M. Altman, J. Bardhan, B. Tidor, and J. White, "Fast methods for simulations of biomolecule electrostatics," in *Proc. IEEE/ACM Int. Conf. Computer Aided-Design*, Nov. 2002, pp. 466–473.

[48] B. Buchmann and J. White, "Revisiting the precorrected FFT method for fast solution of integral equations," , to be published.

[49] C. Tai, *Dyadic Green's Functions in Electromagnetic Theory*. Piscataway, NJ: IEEE Press, 1994.

[50] S. Ramo, J. Whinnery, and T. Duzer, *Fields and Waves in Communication Electronics*. New York: Wiley, 1994.
[51] R. F. Harrington, *Time-Harmonic Electromagnetic Fields*. New York: McGraw-Hill, 1961.
[52] M. Kamon, "Fast parasitic extraction and simulation of three-dimensional interconnect via quasi-static analysis," Ph.D. dissertation, Electr. Eng. Comput. Sci. Dept., Mass. Inst. Technol., Cambridge, MA, 1998.
[53] W. Hackbush, *Integral Equations, Theory, and Numerial Treatment*. Basel, Switzerland: Birkhauser Verlag, 1989.
[54] J. Huang, J. Jia, and J. K. White, "A note on integral equation formulations for impedance extraction in three dimensions," , to be published.
[55] H. Haus and J. Melcher, *Electromagnetic Fields and Energy*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[56] J. L. Hess and A. M. O. Smith, "Calculation of potential flow about arbitrary bodies," *Progress in Aeronautical Science*, pp. 1–138, 1966.
[57] Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," in *SIAM J. Sci. Statist. Comput.*, vol. 7, Jul. 1986, pp. 856–869.
[58] A. Brandt and A. A. Lubrecht, "Multilevel matrix multiplication and fast solution of integral equations," *J. Computat. Phys.*, vol. 90, pp. 348–370, 1990.
[59] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. New York: Adam Hilger, 1988.
[60] T. Sarkar, E. Arvas, and S. Rao, "Application of FFT and the conjugate gradient method for the solution of electromagnetic radiation from electrically large and small conducting bodies," *IEEE Trans. Antennas Propag.*, vol. 34, no. 5, pp. 635–640, May 1986.
[61] E. Bleszynski, M. Bleszynski, and T. Jarozewicz, "AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.*, vol. 31, pp. 1225–1251, Sept.–Oct. 1996.
[62] S. Gedney, A. Zhu, W. Tang, and P. Petre, "High-order precorrected FFT solution for electromagnetic scattering," in *IEEE Antennas and Propagation Society Int. Symp.*, 2003, pp. 566–569.
[63] G. H. Golub and C. F. V. Loan, *Matrix Computation*, 2nd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1989.
[64] F. W. Grover, *Inductance Calculations, Working Formulas, and Tables*. New York: Govt. Print. Off., 1946.



Zhenhai Zhu (S'02–M'04) received the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002 and 2004, respectively.

From 1997 to 1999, he was a Senior Software Engineer at Ultima Interconnect Technology, San Jose, CA. From 2000 to 2004, he was a Research Assistant in the Computational Prototyping Group, Research Laboratory of Electronics, MIT. In 2002, he worked as a summer intern at Hewlett Packard, Marlborough, MA. From 2004 to 2005, he was a Postdoctoral Fellow at IBM T. J. Watson Research Center, Yorktown Heights, NY. He joined Cadence Berkeley Laboratories as a Research Scientist in 2005. His research interests include parasitic extraction, model order reduction, fast integral equation solver (stochastic and deterministic), application of numerical techniques to the analysis of interdisciplinary engineering problems including micro-electro-mechanical systems (MEMS), power electronics, and bio-molecular modeling.

Dr. Zhu is the recipient of 2004 Josef Raviv Memorial Postdoctoral Fellowship in Computer Science and Computer Engineering sponsored by IBM.

Ben Song received the B.S. degree from Nanjing University of Science and Technology, China, in 1994, and the M.S. and Ph.D. degrees from the Department of Electrical and Electronics Engineering, Southeast University, Nanjing, China, in 1997 and 2000, respectively.

From February 2001 to July 2003, he was a Postdoctoral Associate at the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge. He is currently an Engineer with Cadence Design Systems, Inc., San Jose, CA. His research interests include computational electromagnetics and large-scale circuit simulation.

Jacob K. White (S'80–M'83) received the B.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, and the S.M. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley.

He worked at the IBM T. J. Watson Research Center from 1985 to 1987, and was the Analog Devices Career Development Assistant Professor at the Massachusetts Institute of Technology from 1987 to 1989. He is currently with the Massachusetts Institute of Technology where he is a Professor in electrical engineering and computer science and an Associate Director of the Research Laboratory of Electronics. His current research interests are in numerical algorithms for problems in circuits, interconnect, micromachined devices and biological systems.

Dr. White was a 1988 Presidential Young Investigator. He was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN from 1992 to 1996, and was Chair of the International Conference on Computer-Aided Design in 1999.