

Efficient Techniques for 3-D Impedance Extraction Using Mixed Boundary Element Method

Fang Gong^{*}, Wenjian Yu^{*}, Zeyi Wang^{*}, Zhiping Yu⁺, Changhao Yan⁺⁺

^{*}Department of Computer Science,
Tsinghua University,
Beijing 100084, China

⁺Institute of Microelectronics,
Tsinghua University,
Beijing 100084, China

⁺⁺Department of Microelectronics,
Fudan University,
Shanghai 201203, China

Tel:+86-10-62774768, Email:gongf05@mails.tsinghua.edu.cn, {yu-wj,wangzy,yuzhip}@tsinghua.edu.cn, yanch@fudan.edu.cn

Abstract - In this paper, we describe the algorithms implemented in MBEM, a program for wideband impedance extraction of complicated 3-D structures. MBEM is based on a mixed boundary element method (BEM), which reduces the number of unknowns from about $7N$ in FastImp to $4N$, for MQS analysis. Efficient techniques are proposed to handle the extra matrix multiplication, form post-process matrices, and solve the final linear equation system. The inaccuracy of calculation using FastImp at low frequency is also analyzed, which shows the mixed BEM eliminates it completely. Experiments on several typical 3-D structures validate the advantage of MBEM over FastImp, on both accuracy and efficiency.

I Introduction

In the area of interconnect structure modeling and analysis, three-dimensional (3-D) electromagnetic simulation is increasingly significant, but associated with many challenges. For instance, due to the skin and proximity effect at high frequencies, the impedance becomes frequency-dependent, which leads to problems in the simulation and design. Hence, electromagnetic simulation tools should provide accurate and efficient solution across the entire frequency range.

An impedance extractor FastImp was developed and validated to be a fast wideband solver for complicated 3-D structures in [1]. FastImp is based on the surface integral formulation proposed in [2], and involves several efficient computational techniques. Compared with FastHenry [3], which is based on partial equivalent element circuit (PEEC), FastImp avoids discretization of the entire domain of conductors or substrates. However, there are some issues plaguing FastImp. Firstly, impedance extracted by FastImp lacks accuracy at low frequency [4], which is evidenced by experiments. Secondly, the number of discretization unknowns is often too large which is, for example, up to $6N+U$ for magneto-quasistatic (MQS) analysis, where N and U are the number of panels and vertexes, respectively.

Recently, a new formulation called mixed boundary element method (BEM) was proposed for 3-D impedance extraction [5]. This method introduces an indirect boundary integral equation (BIE) to describe the electrical field in the inside domain of conductor, while using the direct BIE for the domain among conductors, as that in FastImp. The mixed BEM reduces the unknown number in the final linear system by substituting the imaginary dipole $\bar{\mu}$ for electric field \bar{E}

and its derivative. For example, it employs only $3N+U$ unknowns for MQS analysis. However, the work in [5] does not result in a practical impedance extractor, because several issues largely hurt the computational efficiency. Firstly, when generating the final linear equation system, some costly extra matrix multiplication for \mathbf{Q}_1 matrix is needed. Secondly, the mixed BEM has to form the post-process matrices, which convert the solved dipole $\bar{\mu}$ to electric field \bar{E} and its derivative $\partial\bar{E}/\partial n$. Lastly, efficient technique for solving the linear equation system was not proposed. With these issues unresolved, the mixed BEM is only established theoretically, but not for practical 3-D impedance extraction.

In this paper, several techniques are proposed to address the above-mentioned issues. Firstly, the precorrected FFT (pFFT) algorithm proposed in [6] is integrated in the mixed BEM. With the help of it, we propose efficient techniques to reduce the computational expense of the extra matrix multiplication and forming the post-process matrices. Then, a new matrix organization is proposed, along with an efficient scaling and preconditioning technique, to facilitate the fast solution of the final linear equation system. We also propose a reuse scheme and other implementation skills, to take advantage of the special structure of overall coefficient matrix. These techniques improve the computational efficiency of mixed BEM greatly. Besides, the low frequency problem of FastImp is analyzed, which indicate that the same problem is completely eliminated in the mixed BEM. With the proposed techniques, a fast impedance extractor named MBEM is developed. Several typical 3-D structures are calculated to demonstrate the efficiency and accuracy of MBEM.

II. Mixed BEM Formulation for Impedance Extraction

FastImp is based on the surface integral formulation proposed in [2], and capable of MQS, electromagneto-quasistatic (EMQS) and even full-wave analysis. The main integral equations inside FastImp are:

$$\frac{1}{2}\bar{E}(y) = \int_{S_j} G_1(y,x) \frac{\partial\bar{E}(x)}{\partial n_x} dx - \int_{S_j} \frac{\partial G_1(y,x)}{\partial n_x} \bar{E}(x) dx, \quad (1)$$

$$-\frac{1}{2}\bar{E}(y) = \int_S G_0(y,x) \frac{\partial\bar{E}(x)}{\partial n_x} dx - \int_S \frac{\partial G_0(y,x)}{\partial n_x} \bar{E}(x) dx + \nabla\varphi(y), \quad (2)$$

where \bar{E} is electric field vector, φ is electric potential. S_j is the j th conductor surface, and S is the summation of all conductor surfaces. G_0 and G_1 are the Green's function $e^{ikr}/4\pi r$, and they are frequency dependent except $G_0 = 1/4\pi r(y,x)$ for MQS or EMQS analysis. $r(y,x)$ denotes the distance between two points y and x .

This work is supported by NSFC under Grant 60401010 and in part by the Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

After decomposing the vector boundary integral equations into scalar boundary integral equations, and applying the collocation scheme based on surface panel discretization, a set of linear equations is obtained. Combining with other boundary conditions and discretizing techniques, overall linear equation system $\mathbf{Ax}=\mathbf{b}$ is formed, where discretized \bar{E} , $\partial\bar{E}/\partial n$, ϕ , and ρ are unknowns.

The major work of forming coefficient matrix \mathbf{A} is calculating the integrals in (1) and (2). Their kernel is of the Green's functions, as listed in (3), where a , b are the global indexes of the panels, and $panel_b$ is the b th panel.

$$\begin{cases} P_0(a,b) = \int_{panel_b} \frac{1}{4\pi r(y_a,x)} dx \\ D_0(a,b) = \int_{panel_b} \frac{\partial}{\partial n_x} \left[\frac{1}{4\pi r(y_a,x)} \right] dx \\ P_1(a,b) = \int_{panel_b} G_1(y_a,x) dx = \int_{panel_b} \frac{e^{ikr(y_a,x)}}{4\pi r(y_a,x)} dx \\ D_1(a,b) = \int_{panel_b} \frac{\partial G_1(y_a,x)}{\partial n_x} dx = \int_{panel_b} \frac{\partial}{\partial n_x} \left[\frac{e^{ikr(y_a,x)}}{4\pi r(y_a,x)} \right] dx \end{cases} \quad (3)$$

The first step of the mixed BEM is to transfer (1) into a form of indirect boundary integral equation (BIE). Eq. (1) corresponds to a vector Helmholtz equation of \bar{E} , which can be decomposed into three scalar Helmholtz equations. Without loss of generality, taking one of them:

$$\nabla^2 E_x - i\omega\mu\sigma E_x = 0 \quad , \quad (4)$$

for example (E_x denotes the E component along \hat{x} direction), one can transform it into an indirect BIE according to the potential theory [7]:

$$E_x(y) = \int_{\Gamma_i} \frac{\partial G_1(y,x)}{\partial n(x)} \mu_x(x) dS_x \quad . \quad (5)$$

Here $\mu_x(x)$ is an imaginary dipole distribution imposed on Γ_i , with the dipole axis along the out normal direction of boundary. So, Eq. (5) reflects the relationship between E_x and any evaluation point y on the conductor boundary Γ_i .

Furthermore, if assuming that $E_x(y)$ is differentiable at point y , one can obtain:

$$\frac{\partial E_x(y)}{\partial n(y)} = \frac{\partial}{\partial n(y)} \left[\int_{\Gamma_i} \frac{\partial G_1(y,x)}{\partial n(x)} \mu_x(x) dS_x \right] \quad . \quad (6)$$

After obtaining the indirect BIE forms of \bar{E} and $\partial\bar{E}/\partial n$, substituting them into the boundary integral equations employed by FastImp except (1), one can get a set of mixed BIEs. After surface discretization, the final linear equation system can be generated as well. Taking the MQS analysis as an example, the overall linear equation system became:

$$\begin{pmatrix} T_{1x}\mathbf{Q}_1 & T_{1y}\mathbf{Q}_1 & T_{1z}\mathbf{Q}_1 & A_{11} \\ T_{2x}\mathbf{Q}_1 & T_{2y}\mathbf{Q}_1 & T_{2z}\mathbf{Q}_1 & A_{12} \\ N_{ncx}\mathbf{D}_1 & N_{ncy}\mathbf{D}_1 & N_{ncz}\mathbf{D}_1 & 0 \\ T_{1x}\mathbf{D}_1 & T_{1y}\mathbf{D}_1 & T_{1z}\mathbf{D}_1 & 0 \\ T_{2x}\mathbf{D}_1 & T_{2y}\mathbf{D}_1 & T_{2z}\mathbf{D}_1 & 0 \\ N_{cx}\mathbf{D}_2 & N_{cy}\mathbf{D}_2 & N_{cz}\mathbf{D}_2 & 0 \\ \mathbf{Q}_{2x} & \mathbf{Q}_{2y} & \mathbf{Q}_{2z} & 0 \\ 0 & 0 & 0 & \mathbf{I}_c \end{pmatrix} \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \\ \phi \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{bmatrix} \quad , \quad (7)$$

$$\mathbf{Q}_1 = \mathbf{P}_0\mathbf{D}_2 - \mathbf{D}_0\mathbf{D}_1 \quad ,$$

$$\mathbf{Q}_{2\alpha} = \mathbf{C}_{d\alpha}\mathbf{D}_2 + \mathbf{C}_\alpha\mathbf{D}_1, \quad \alpha = x, y, z,$$

And the entry of matrix \mathbf{D}_2 is:

$$D_2(a,b) = \frac{\partial}{\partial n(y_a)} \int_{panel_b} \frac{\partial}{\partial n_x} \left[\frac{e^{-ikr(y_a,x)}}{4\pi r(y_a,x)} \right] dx \quad . \quad (8)$$

In practice, matrices \mathbf{P}_0 , \mathbf{D}_0 and \mathbf{D}_1 are the same as those in FastImp. The difference only lies in replacing \mathbf{P}_1 with \mathbf{D}_2 , whose element is calculated with (8). The mixed BEM reduces the unknowns in final equation system (7) by replacing \bar{E} and its derivative with $\bar{\mu}$. But generating and solving the system remains a challenging task.

III. Efficient Techniques for the Mixed BEM Solver

To make the mixed BEM into a practical and fast impedance solver, pFFT algorithm is employed to accelerate matrix-vector multiplication. Several efficient techniques are also proposed. Firstly, two remaining problems of the mixed BEM in [5] are discussed and resolved. Then, efficient organizing, scaling and preconditioning techniques are proposed for the linear equation system of mixed BEM. Lastly, we give implementation skills and the overall algorithm flow.

A. Extra Matrix Multiplication and Post-Process Matrices

In the linear equation system (7), matrix \mathbf{P}_0 , \mathbf{D}_0 , \mathbf{D}_1 and \mathbf{D}_2 are filled with boundary integrals, while other matrices are sparse matrices. Generating these integral-related matrices costs a lot of time. Compared with the formation of linear equation system in FastImp, the extra calculation in the mixed BEM is mainly that for matrix \mathbf{Q}_1 , which involves multiplication of the integral-related matrices. If conventional approach is used, the construction of matrix \mathbf{Q}_1 has a large computational complexity of $O(n^3)$, where n is the number of surface panels. This is obviously inhibitive.

To revamp this problem, the pFFT algorithm can be taken advantage of. The pFFT algorithm was invented to accelerate the product of integral-related matrix and vector, with the computational complexity reduced from $O(n^2)$ to $O(n)$ [6]. And, since the iterative Krylov subspace solver is usually employed to solve the linear equation system, it is not necessary to generate (7) explicitly. Actually, we just need to know the matrix-vector product related with \mathbf{Q}_1 and other matrix blocks in (7). The real problem of the extra matrix \mathbf{Q}_1 becomes how to calculate the related matrix-vector product efficiently. Taking the operation $\mathbf{T}_{1x}\mathbf{Q}_1\mu_x$ in (7) for example, we have the following equality:

$$\mathbf{T}_{1x}\mathbf{Q}_1\mu_x = \mathbf{T}_{1x}(\mathbf{P}_0\mathbf{D}_2 - \mathbf{D}_0\mathbf{D}_1)\mu_x = \mathbf{T}_{1x}[\mathbf{P}_0(\mathbf{D}_2\mu_x) - \mathbf{D}_0(\mathbf{D}_1\mu_x)] \quad . \quad (9)$$

Now, we just need to calculate four products of integral-related matrix and vector, and each product is performed with the pFFT algorithm. With this technique, the computation for the extra matrix \mathbf{Q}_1 is greatly reduced.

The other flaw of the mixed BEM is about the post-process matrices. Since the electric field \bar{E} and its derivative $\partial\bar{E}/\partial n$ are needed to calculate the port current for impedance extraction, we have to convert the solved unknown $\bar{\mu}$ to \bar{E} and $\partial\bar{E}/\partial n$ with equations like (5) and (6). This involves the

calculation of matrix D_1 and D_2 . But the coefficient matrix of (7) is not formed explicitly because the pFFT algorithm is used, which means that we can not access the D_1 and D_2 directly.

After careful consideration, we find out that not the whole matrix is needed for the post-process; only the rows in the matrix corresponding to contact panels are needed, whose inner products with $\bar{\mu}$ solutions are utilized for current calculation. Therefore, our idea is to generate the related rows of D_1 and D_2 explicitly. Since the contact panels usually account for a small portion of the whole surface panels, calculating the integrals for the contact-related rows of D_1 and D_2 is very fast. And, if there are N_c conductors, these entries are used for N_c times for extracting the whole impedance matrix. This technique greatly reduces the computing time and memory usage for the post-process matrices.

B. Matrix Organization and Scaling

The linear equation system in (7) is solved with Krylov subspace iterative solver. The convergence rate of the solver is related with the distribution of nonzeros in the coefficient matrix. Below we propose reorganization and scaling technique for the coefficient matrix. To clarify the nonzero distribution of the coefficient matrix in the mixed BEM, an example with only one straight wire is used here. Fig. 1(a) shows the matrix population of (7) for this example. It is obvious that many zero entries locate on or near the main diagonal, which is harmful to the iterative convergence. Thus, appropriate manipulation of rows and columns in (7) is performed to disperse nonzero entries at or near the diagonal. We exchange the second row with the seventh row, and exchange the first column with the second column. The resulting equation formulation is shown in (10). Now, the nonzero distribution for the example is shown in Fig. 1(b).

$$\begin{pmatrix} T_{1y}Q_1 & T_{1x}Q_1 & T_{1z}Q_1 & A_{r1} \\ Q_{2y} & Q_{2x} & Q_{2z} & 0 \\ N_{ncy}D_1 & N_{ncx}D_1 & N_{ncz}D_1 & 0 \\ T_{1y}D_1 & T_{1x}D_1 & T_{1z}D_1 & 0 \\ T_{2y}D_1 & T_{2x}D_1 & T_{2z}D_1 & 0 \\ N_{cy}D_2 & N_{cx}D_2 & N_{cz}D_2 & 0 \\ T_{2y}Q_1 & T_{2x}Q_1 & T_{2z}Q_1 & A_{r2} \\ 0 & 0 & 0 & I_c \end{pmatrix} \begin{bmatrix} \mu_y \\ \mu_x \\ \mu_z \\ \varphi \end{bmatrix} = \begin{bmatrix} b_1 \\ b_7 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_2 \\ b_8 \end{bmatrix} \quad (10)$$

Another useful technique is scaling. Suppose the average size of panels is $O(u)$, we can roughly estimate the scale of the coefficients in the final linear equation system. Since points y_a and x are on the conductor surface, $1/4\pi r(y_a, x)$ and $e^{ikr(y_a, x)}/4\pi r(y_a, x)$ are $O(1/u)$, and dx is $O(u^2)$. Hence, $P_0(a, b)$ is $O(u)$. Similarly, $D_0(a, b)$ is $O(1)$, as are the entries in matrix

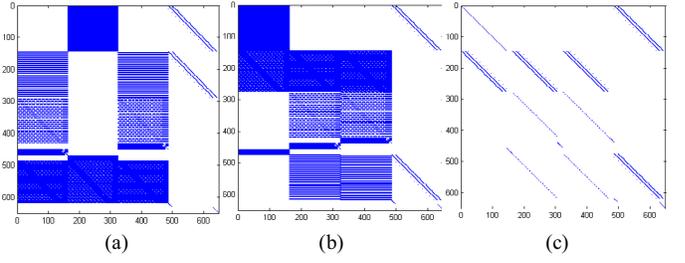


Fig. 1. The nonzero distribution of: (a) the coefficient matrix without matrix organization, (b) the coefficient matrix with matrix organization, and (c) the D_1 and D_2 . Therefore, entries in matrix Q_1 is $O(1)$. The elements in C_{dx} , C_{dy} and C_{dz} is $O(u^2)$, and entries in C_x , C_y and C_z are $O(u)$. Thus elements in Q_{2x} , Q_{2y} and Q_{2z} are $O(u)$. Above analysis implies a simple scaling strategy: to scale the second row in (10) with $1/u$ to enforce all matrix blocks to be $O(1)$. The resulting equation system becomes:

$$\begin{pmatrix} T_{1y}Q_1 & T_{1x}Q_1 & T_{1z}Q_1 & A_{r1} \\ \frac{1}{u}Q_{2y} & \frac{1}{u}Q_{2x} & \frac{1}{u}Q_{2z} & 0 \\ N_{ncy}D_1 & N_{ncx}D_1 & N_{ncz}D_1 & 0 \\ T_{1y}D_1 & T_{1x}D_1 & T_{1z}D_1 & 0 \\ T_{2y}D_1 & T_{2x}D_1 & T_{2z}D_1 & 0 \\ N_{cy}D_2 & N_{cx}D_2 & N_{cz}D_2 & 0 \\ T_{2y}Q_1 & T_{2x}Q_1 & T_{2z}Q_1 & A_{r2} \\ 0 & 0 & 0 & I_c \end{pmatrix} \begin{bmatrix} \mu_y \\ \mu_x \\ \mu_z \\ \varphi \end{bmatrix} = \begin{bmatrix} b_1 \\ \frac{1}{u}b_7 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_2 \\ b_8 \end{bmatrix} \quad (11)$$

This scaling technique largely reduces the condition number of the coefficient matrix, and also brings benefit to the convergence rate.

To verify the effect of the reorganizing and scaling technique, the single wire example and another cross-over structure with four conductors are calculated. The GMRES iteration numbers for the equation with and without these techniques are listed in Table I. From the table, we can see that the iteration number is largely reduced. Note that a same strategy of preconditioning is used for both situations.

TABLE I
THE NUMBER OF GMRES ITERATION WITH OR WITHOUT ADJUSTMENT

	Number of Iteration in GMRES	
	Without Adjustment	With Adjustment
Wire	9	7
Bus2x2	22	13

C. Preconditioning Technique

According to the proposed adjustment technique for the coefficient matrix, we propose a preconditioner matrix which is expressed as (12). The superscript D indicates that the matrix block is just the diagonal part of corresponding matrix. Fig. 1 (c) shows the nonzero distribution of the preconditioner

TABLE II

THE PRECONDITIONER CONSTRUCTION TIME AND ITERATION NUMBER IN MBEM AND FASTIMP FOR DIFFERENT STRUCTURES

	Number of Unknowns		Construction (Second)		LU Factorization (Second)		Number of Iteration*	
	FastImp	MBEM	FastImp	MBEM	FastImp	MBEM	FastImp	MBEM
Wire	1136	650	0.04	0.19	0.01	0.07	9	8
Bus1x1	2272	1300	0.08	0.74	0.01	0.42	18	8.5
Bus2x2	3536	2024	0.16	1.22	0.02	1.24	21	9
Spiral	3418	1954	0.2	1.38	0.01	0.32	10	8

*The value is the average number of iteration for one right-hand side when the structure involves more than one conductor.

matrix for the single wire example.

$$\begin{pmatrix} T_{1y} \mathbf{Q}_1^D & T_{1x} \mathbf{Q}_1^D & T_{1z} \mathbf{Q}_1^D & A_{t1} \\ \frac{1}{u} \mathbf{Q}'_{2y} & \frac{1}{u} \mathbf{Q}'_{2x} & \frac{1}{u} \mathbf{Q}'_{2z} & 0 \\ N_{ncy} \mathbf{D}_1^D & N_{ncx} \mathbf{D}_1^D & N_{ncz} \mathbf{D}_1^D & 0 \\ T_{1y} \mathbf{D}_1^D & T_{1x} \mathbf{D}_1^D & T_{1z} \mathbf{D}_1^D & 0 \\ T_{2y} \mathbf{D}_1^D & T_{2x} \mathbf{D}_1^D & T_{2z} \mathbf{D}_1^D & 0 \\ N_{cy} \mathbf{D}_2^D & N_{cx} \mathbf{D}_2^D & N_{cz} \mathbf{D}_2^D & 0 \\ T_{2y} \mathbf{Q}_1^D & T_{2x} \mathbf{Q}_1^D & T_{2z} \mathbf{Q}_1^D & A_{t2} \\ 0 & 0 & 0 & I_c \end{pmatrix} \quad (12)$$

$$\mathbf{Q}_1^D = (\mathbf{P}_0 \mathbf{D}_2^D - \mathbf{D}_0 \mathbf{D}_1^D)^D$$

$$\mathbf{Q}_{2\alpha} = \mathbf{C}_{d\alpha} \mathbf{D}_2^D + \mathbf{C}_\alpha \mathbf{D}_1^D, \quad \alpha = x, y, z$$

Extensive experiments have demonstrated that this preconditioning technique not only costs little time for construction, but is also robust and efficient. The comparison between MBEM and FastImp on the preconditioner construction time and efficiency is presented in Table II. Both methods use a similar strategy of precondition, but Table II indicates that the preconditioner construction time for our method is larger than that for FastImp. This is due to the larger complexity of calculating entries of \mathbf{Q}_1 , $\mathbf{Q}_{2\alpha}$ ($\alpha=x, y, z$), and \mathbf{D}_2 in MBEM.

D. Implementation Optimization

1. The special structure of the matrix blocks in coefficient matrix \mathbf{A} can be utilized to reduce the CPU time by tabulating computed products and reusing them. In detail, the method for the extra matrix multiplication (i.e. Eq. (9)) can be further simplified by calculating only six products $\mathbf{D}_1 \boldsymbol{\mu}$, $\mathbf{D}_2 \boldsymbol{\mu}$ ($\boldsymbol{\mu}=\boldsymbol{\mu}_x, \boldsymbol{\mu}_y, \boldsymbol{\mu}_z$) and reusing them:

$$\begin{aligned} \mathbf{Q}_1 \boldsymbol{\mu}_\alpha &= (\mathbf{P}_0 \mathbf{D}_2 - \mathbf{D}_0 \mathbf{D}_1) \boldsymbol{\mu}_\alpha = [\mathbf{P}_0 (\mathbf{D}_2 \boldsymbol{\mu}_\alpha) - \mathbf{D}_0 (\mathbf{D}_1 \boldsymbol{\mu}_\alpha)] \\ \mathbf{Q}_{2\alpha} \boldsymbol{\mu}_\beta &= [\mathbf{C}_{d\alpha} \mathbf{D}_2 + \mathbf{C}_\alpha \mathbf{D}_1] \boldsymbol{\mu}_\beta = [\mathbf{C}_{d\alpha} (\mathbf{D}_2 \boldsymbol{\mu}_\beta) + \mathbf{C}_\alpha (\mathbf{D}_1 \boldsymbol{\mu}_\beta)] \end{aligned} \quad (13)$$

$\alpha = x, y, z; \quad \beta = x, y, z$

For the single wire example, this reuse scheme reduces the number of products from 42 to 12, and about 71.4% of the CPU time for $\mathbf{A}\mathbf{x}$ product is saved.

2. The equation for inside domain of each conductor (1) is replaced with indirect BIEs, like (5) and (6). Forming these discretized indirect BIEs is actually to form matrix \mathbf{D}_1 and \mathbf{D}_2 . Since in each discretized indirect BIE, all discretized variables are on the boundary of one conductor, there is no coupling between variables belonging to different conductors. Thus, matrix \mathbf{D}_1 and \mathbf{D}_2 are actually block diagonal matrices (as shown in Fig. 2). Furthermore, if two conductors have the same type, geometry and discretization, their corresponding

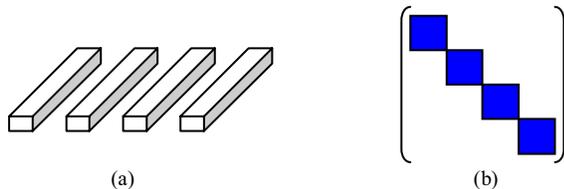


Fig. 2. (a) A problem with four conductors, and (b) the corresponding population of matrix \mathbf{D}_1 or \mathbf{D}_2 .

matrix blocks are just the same. This enables a reuse scheme for constructing \mathbf{D}_1 and \mathbf{D}_2 matrices with the pFFT algorithm. This technique utilizes the regularity of structure and can achieve large reduction in computing time.

3. Since \mathbf{Q}_1^D in the preconditioner only involves its diagonal elements, it is feasible to accelerate the formation of \mathbf{Q}_1^D . With the equality shown in (14), the computational complexity is greatly reduced.

$$\mathbf{Q}_1^D = (\mathbf{P}_0 \mathbf{D}_2^D - \mathbf{D}_0 \mathbf{D}_1^D)^D = \mathbf{P}_0^D \mathbf{D}_2^D - \mathbf{D}_0^D \mathbf{D}_1^D \quad (14)$$

$$\text{i.e., } Q_1(j, j) = P_0(j, j) D_2(j, j) - D_0(j, j) D_1(j, j)$$

E. Algorithm Flow

For the purpose of clarity, below we give the main steps in the overall algorithm flow.

Step 1: Fill the \mathbf{P} , \mathbf{I} , \mathbf{D} , \mathbf{H} matrices in the pFFT algorithm for the integral-related matrices \mathbf{P}_0 , \mathbf{D}_0 , \mathbf{D}_1 , \mathbf{D}_2 .

Step 2: Calculate the rows of \mathbf{D}_1 and \mathbf{D}_2 corresponding to contact panels, and store them for the post-process.

Step 3: Construct the preconditioner matrix, according to the formula given in Section III.C.

Step 4: The final linear equation system is organized and scaled with the technique in Section III.B, and then solved by the preconditioned GMRES solver. In each GMRES iteration, the extra matrix multiplications are handled with the technique given in Section III.A.

Step 5: Calculate the port current and thus impedance with the solution and post-process matrices.

IV. Discussion about the Low-Frequency Problem

The inaccuracy of FastImp at low frequencies is due to its inside boundary integral formulation, i.e. that proposed in [2] and [8]. Ref. [8] has pointed out that this is caused by a $\sqrt{\omega}$ term in the Taylor's expansion of the kernel of Green's function \mathbf{P}_1 (see (3)). Although in theory it should have no impact, the error induced by this reason becomes significant due to numerical issues. The Taylor's expansion of the integral kernel is:

$$\frac{e^{ikr}}{r} = \frac{1}{r} + ik + \frac{1}{r} \frac{(ik)^2 r^2}{2} + O(\omega), \quad (15)$$

where k is $O(\sqrt{\omega})$ and the second term ik is $O(\sqrt{\omega})$. The $\sqrt{\omega}$ term can dominate the ω at low frequencies, so as to cause larger error.

In [8], it is also claimed that the dipole kernel expansion $\partial(e^{ikr}/r)/\partial n$ does not include the $\sqrt{\omega}$ term. This is because:

$$\begin{aligned} \frac{\partial}{\partial n} \frac{e^{ikr}}{r} &\approx \frac{ikr-1}{r^2} e^{ikr} \frac{\partial r}{\partial n} \approx \frac{ikr-1}{r^2} \left(1 + ikr + \frac{(ikr)^2}{2} \right) \frac{\partial r}{\partial n} \\ &\approx - \left(\frac{1+k^2 r^2}{r^2} + \frac{1-ikr}{r^2} \frac{(ikr)^2}{2} \right) \frac{\partial r}{\partial n} \\ &\approx - \frac{1}{r^2} \frac{\partial r}{\partial n} - \frac{k^2}{2} \frac{\partial r}{\partial n} \end{aligned} \quad (16)$$

$$\approx \frac{\partial}{\partial n} \left(\frac{1}{r} \right) - \frac{k^2}{2} \frac{\partial r}{\partial n}$$

where the $\sqrt{\omega}$ term is totally eliminated.

The dipole kernel is just the kernel of the integral \mathbf{D}_1 in (3). In the mixed BEM, four kinds of integrals are calculated, i.e. \mathbf{P}_0 , \mathbf{D}_0 , \mathbf{D}_1 in (3) and \mathbf{D}_2 in (8). The kernels of \mathbf{P}_0 and \mathbf{D}_0 are frequency-independent, so they have no relation with the low-frequency problem. And with a similar deduction like (15) we know that the kernel of \mathbf{D}_2 does not involve the $\sqrt{\omega}$ term either. That is, no integral kernel in the mixed BEM is disturbed by the $\sqrt{\omega}$ term, and at low frequencies the mixed BEM shall have no numerical problem. This conclusion is verified by the accuracy of MBEM for extraction at low frequencies.

V. Numerical Results

Based on the proposed algorithms, we have developed a program with name of MBEM for impedance extraction. In this section, two small examples are firstly used to validate MBEM's accuracy. Then, some large examples demonstrate its speed. All experiments are carried out on a Linux sever with 3GHz CPU, and the MQS analysis mode is assumed.

A. Accuracy Validation

The first example is a single wire with the length of $4\mu\text{m}$ and a $1\mu\text{m} \times 1\mu\text{m}$ cross section. The structure and surface discretization are shown in Fig. 3(a). The impedance is

extracted for a frequency range from 10^4 Hz to 10^{11} Hz.

For this example, the computational results are compared with FastImp. Both FastImp [9] and MBEM employ a uniform $3 \times 3 \times 12$ discretization. We also use FastHenry to give a reference result. In FastHenry, 288 filaments are partitioned to capture the skin effect at the highest frequency. The resistance results from these programs are shown in Fig. 3(b), where the curves from three programs coincide with each other. The largest discrepancy occurs at 100 GHz, where the result of FastHenry is about 2.7% smaller.

Fig. 3(c) shows the comparison of three programs for wire inductance. It clearly reveals the low frequency problem of FastImp; the result from FastImp at 10 KHz is at most 34% smaller than that from FastHenry at 10 KHz. On the contrary, the results from MBEM match those from FastHenry very well, with the discrepancy less than 3% for the whole frequency range. It is noticeable that all methods show an inductance drop at high frequencies. And the discrepancy among them becomes negligible at high frequencies.

The second example is the "bus2x2" case got from FastImp's package. Its structure and geometries are shown in Fig. 4(a). With this example we verify that MBEM can calculate the mutual impedance accurately, which involves the proximity effect at high frequency. The computational results of mutual resistance and inductance of parallel wires are shown in Fig. 4(b) and Fig. 4(c).

As shown in Fig. 4(b), FastImp and MBEM correctly capture the non-zero mutual resistance at high frequency,

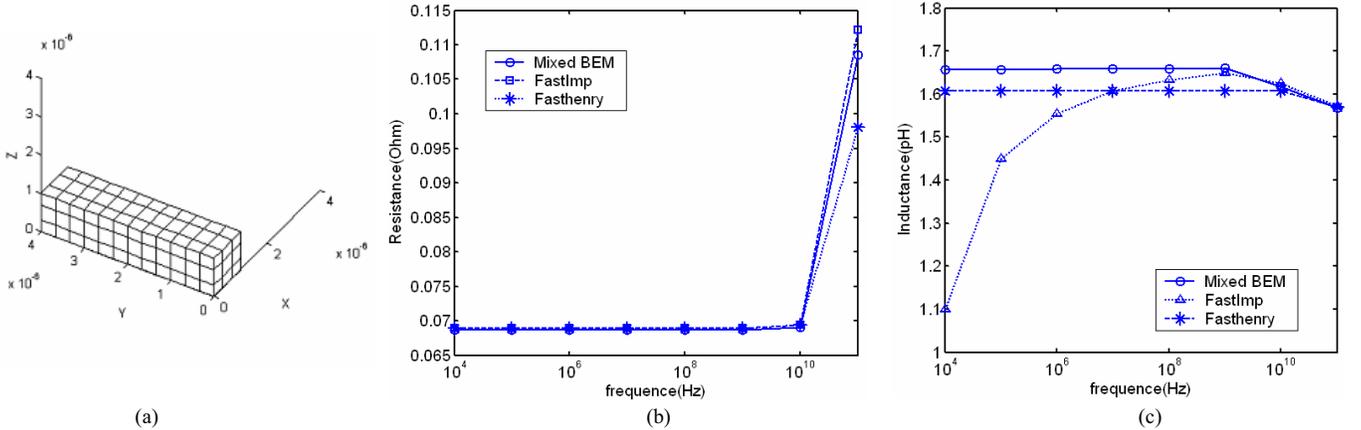


Fig. 3. (a) A wire structure with panel discretization. (b) Resistance of the wire, and (c) Inductance of the wire. The results are compared among FastImp, MBEM and FastHenry.

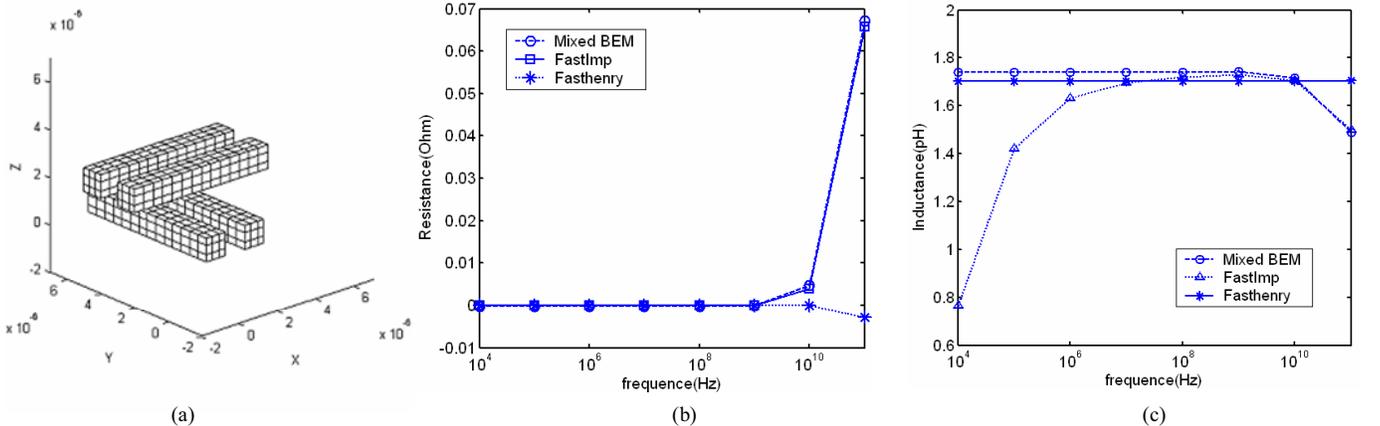


Fig. 4. (a) A bus2x2 structure with panel discretization. (b) Mutual resistance of the bus2x2, and (c) the mutual inductance. The results are compared among FastImp, MBEM and FastHenry.

where the result from FastHenry is unreasonable. Fig. 4(c) demonstrates again the low frequency problem of FastImp, while the results from FastHenry and MBEM coincide with each other very well at low frequencies. But at high frequencies, it is clear that the results from FastImp and MBEM are accurate which reflect the inductance variation due to high frequency effects. At the same time, the result from FastHenry yields a maximum error of 16% when frequency is 100GHz. In fact, the result of FastHenry with finer discretization does not make a difference, which suggests that FastHenry is not suitable for the extraction of high-frequency mutual inductance.

These two examples demonstrate that MBEM does not have the inaccuracy problem at low frequencies, and it has high accuracy for the wide frequency range. The CPU times of FastImp for both examples are 7.2s and 59s respectively, while the times of MBEM are 10.3s and 34.9s. MBEM shows the same order or faster speed than FastImp.

B. Speed Validation

The first large structure is got from [1], which includes five circular spiral inductors of various sizes and number of turns, and two 3-D interconnect structures; The other one is a THU logo, which is comprised of 24 two-turn circular spirals (as shown in Fig. 5). At frequency of 1 GHz, the whole impedance matrix is extracted for each structure.

The discretization numbers and detailed breakdown of CPU time for both examples are shown in Table III and IV, respectively. Table III shows that MBEM has only about half

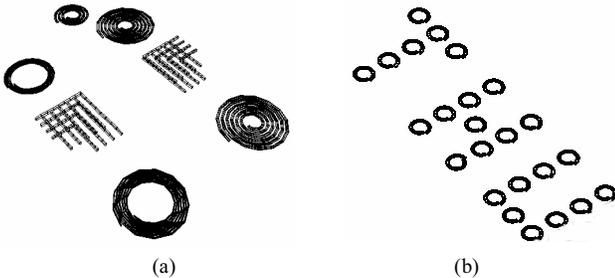


Fig. 5. Two large structures: (a) a portion of RF circuit, and (b) a THU logo.

of the unknowns of FastImp, which in turn causes fewer iterative steps in GMRES solution (see Table IV), and makes MBEM to be faster than FastImp.

It is worth noting that MBEM has to fill post-process matrices, which is an additional step which FastImp does not need. But the reduction of time for equation solution overwhelms the time for filling the post-process matrices. And, the computational time of MBEM for other steps becomes less than or comparable to that of FastImp due to the proposed techniques. Finally, the total computational time of MBEM is at least 42% less than that of FastImp. The memory usage of FastImp is 173 MB and 474 MB for both examples, while that of MBEM is 192 MB and 501 MB, respectively. The little

TABLE III
DISCRETIZATION OF TWO LARGE 3-D STRUCTURES

		RF circuit	THU logo
Number of panels		6,364	17,472
Number of vertices		6,426	17,520
Number of unknowns	FastImp	44,610	122,352
	MBEM	25,518	69,936

TABLE IV

A DETAILED BREAKDOWN OF THE CPU TIME USED BY TWO EXAMPLES.

UNIT IS SECOND

	RF circuit		THU logo	
	FastImp	MBEM	FastImp	MBEM
Fill P, I, D, H matrices	335.0	263.2	811.9	106.2
Form post-process matrices	--	32.5	--	56.9
Form the preconditioner P_r	2.0	4.2	7.8	13.0
LU factorization of P_r	0.16	18.6	0.5	60.0
Product of Ax	7.0	6.9	18.4	15.0
GMRES (tol = 1e-3)	7558 (29 iter)	3011 (11 iter)	7265 (15 iter)	4280 (8 iter)
Total time	7893 (2.19 hour)	3447 (0.96hour)	7919 (2.2hour)	4594 (1.27hour)

more memory spent by MBEM is for storing the post-process matrices.

VI. CONCLUSION

Based on the mixed BEM formulation, a new 3-D impedance extractor MBEM is developed. MBEM employs the pFFT algorithm to extend its scalability. The efficient techniques are proposed to handle the problems of mixed BEM, i.e. those related with the extra matrix multiplication and post-process matrices. Organizing, scaling and preconditioning techniques are proposed to form a fast GMRES equation solver for the linear equation system generated by mixed BEM. The analysis of the low-frequency problem of FastImp shows that the mixed BEM is guaranteed to be accurate and robust across the whole frequency range. The accuracy and efficiency of MBEM has been demonstrated by a variety of examples, which shows its advantage over FastImp for large structures.

References

- [1] Z. Zhu, B. Song, and J. White, "Algorithms in FastImp: a fast and wide-band impedance extraction program for complicated 3-D geometries," *IEEE Trans. CAD*, vol. 24, no. 7, pp. 981-998, 2005.
- [2] J. Wang, J. White, "A wide frequency range surface integral formulation for 3-D RLC extraction," in *Proc. ICCAD*, pp. 453-457, 1999.
- [3] M. Kamon, M. J. Tsuk, and J. White, "FastHenry: a multipole-accelerated 3-D inductance extraction program," *IEEE Trans. MTT*, vol. 42, no. 9, pp. 1750-1758, Sep. 1994.
- [4] X. Hu, L. Daniel, and J. White, "Partitioned conduction modes in surface integral equation-based impedance extraction," in *Proc. PEPE*, pp. 355-358, Oct 2003.
- [5] C. Yan, W. Yu, Z. Wang, "A mixed boundary element method for extracting frequency-dependent inductances of 3D interconnects," in *Proc. ISQED*, San Jose, CA, pp. 709-716, 2006.
- [6] J. R. Phillips and J. White, "A precorrected-FFT method for electrostatic analysis of complicated 3D structures," *IEEE Trans. CAD*, vol. 16, no. 10, pp. 1059-1072, Oct. 1997.
- [7] G. Athanasiadis, "Numerical investigations of direct and indirect integral equations for solving the heat conduction problem," *Computer Methods in Applied Mechanics and Engineering*, vol. 49, no. 22, pp. 203-220, 1985.
- [8] J. Wang, "A new surface integral formulation of EMQS impedance extraction for 3-D structures," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1999.
- [9] Z. Zhu, B. Song, and J. White, FastImp code package. Available: http://www.rle.mit.edu/cpg/research_codes.htm