Enhanced quasi-multiple medium technology for fast finite-domain electrostatic BEM simulation

W. Yu & Z. Wang
Department of Computer Science and Technology, Tsinghua University, China.

Abstract

Fast 3D electrostatic simulations are of increasing importance in the area of VLSI interconnects and MEMS, especially for the current deep submicron semiconductor technology. The parasitic capacitance among the interconnects is usually simulated within a finite domain with the mixed Neumann and Dirichlet boundaries. The boundary element method (BEM) is very suitable for such kind of electrostatic computation. Furthermore, a new technology called quasi-multiple medium (QMM) method has been proposed to accelerate the large-scale BEM computation. The QMM technology has been applied to the 3D finite-domain BEM simulation of the interconnect capacitor, and it greatly reduced the computational time and memory usage. In this paper, an approach is presented to enhance the QMM technology, in which the QMM cutting number is automatically selected before decomposing original medium regions into more fictitious medium blocks. With this approach, a reasonable parameter \((m, n)\) is found to make the QMM technology achieve higher computational speed. Besides, the assembly of the global system of linear equations and a new preconditioner for the GMRES solution of the global linear system are introduced. The enhanced QMM technology makes the 3D finite-domain electrostatic computation much faster, while preserving high accuracy.

1 Introduction

The boundary element method (BEM) is very suitable for the electrostatic simulation, and a quasi-multiple medium (QMM) method has been proposed to
accelerate the large-scale finite-domain BEM computation [1-3]. For a single-medium model problem, the QMM accelerated BEM was analyzed to have $O(N)$ computational complexity [1]. The QMM technology has been applied to the 3D simulation of the interconnect capacitor, and greatly reduced the computational time and memory usage [2-3]. The main idea of QMM technology is to decompose the original medium region into some fictitious medium blocks. However, in the existing applications, the QMM cutting number which determines the QMM decomposition is manually specified or generated by an empirical formula. The QMM cutting number generated by user’s experience usually cannot give QMM good enough performance for generalized cases. In this paper, an SMZ approach to automatically selecting the QMM cutting number is proposed to further reduce the computational time of the QMM based electrostatic simulation. Besides, the assembly of the global system of linear equations and a new preconditioner for the GMRES solution of the system are also presented. Numerical results and comparisons with other existing methods demonstrate the efficiency of the enhanced QMM technology, which makes the 3D finite-domain electrostatic computation much faster, with the accuracy preserved and the adaptability extended.

2 The large-scale electrostatic BEM simulation with the quasi-multiple medium acceleration

2.1 Boundary element method for 3D large-scale electrostatic computation

For a simulation domain involving multiple homogeneous regions, the electrostatic Laplace equation for each region is transformed into the direct boundary integral equation (BIE) by the Green’s formula, respectively. For the homogeneous region $Q_i$, the electrical potential $u$ and flux density $q$ on the boundary surface fulfill the following BIE [4,5]:

$$c_s u_s + \int_{\partial Q_i} q^* u' d\Gamma = \int_{\partial Q_i} u^* q d\Gamma$$

(1)

where $u_s$ is the electrical potential at collocation point $s$, $c_s$ is a constant dependent on boundary geometry near to the point $s$. $u^*$ stands for the fundamental solution, and $q^*$ is the derivative of $u^*$ along the outward normal direction of boundary $\partial Q_i$.

Employing constant quadrilateral elements (All can be regarded as of shape trapezoid after partition), and evaluating the direct BIE at collocation points, one for an element, the discretized BIEs for the $i$th region are got:

$$c_k u_k + \sum_{j=1}^{N_i} (\int_{\Gamma_j} q_{ij}^* d\Gamma) u_j = \sum_{j=1}^{N_i} (\int_{\Gamma_j} u_{ij}^* d\Gamma) q_j, \quad (k=1, \ldots, N_i)$$

(2)

where $N_i$ is the number of the boundary elements in dielectric $i$, and $\Gamma_j$ is the $j$th element. The evaluation of integrals in (2) is the most time-consuming part of
boundary element algorithms, in particular for 3D analysis [6]. They can be classified as the singular integrals and non-singular integrals. For the singular integral, the analytical integral method adopting local polar coordinates is effective [7]. The Gauss-Legendre integration scheme with adaptive determination of integration points is employed to calculate the non-singular integral [6]. But for the nearly singular integrals, when the collocation point is close to the element where the integral is taken, the order of Gauss-Legendre integration is still very high. A semi-analytical method is used here to deal with these nearly singular integrals. With application of the primitive function, the 2D integral taken on trapezoid element is converted into one-dimensional Gauss-Legendre integration. With this semi-analytical integration scheme, the number of the integration points is reduced drastically so that higher computational speed and accuracy are achieved. For more detail of the semi-analytical integration, please refer to [3].

Besides, the $u$ and $q$ fulfill the compatibility equations along the interface of two adjacent regions $a$ and $b$:

\[
\begin{align*}
\varepsilon_a \cdot \partial u_a / \partial n_a &= -\varepsilon_b \cdot \partial u_b / \partial n_b \\
u_a &= u_b
\end{align*}
\]  

(3)

where $\varepsilon_a$ and $\varepsilon_b$ stand for the permittivities of dielectric region $a$ and $b$. With (3), the discretized BIEs for each region can be coupled to get a global linear system. Substituting the Dirichlet and Neumann boundary conditions, the global linear system can be reordered to become:

\[
Ax = f,
\]

(4)

where $x$ is a vector comprising all discretized unknowns of $u$ or $q$. The coefficient matrix $A$ is a large non-symmetric one for 3D problem. A GMRES algorithm [8] with a preconditioner presented later is used to solve the linear system.

2.2 Principle of the quasi-multiple medium method [3]

With formula (2), we can see that in each discretized BIE the discretized variables are within the same region as the collocation point. So, in the global linear system (4) for a multi-region structure, every non-zero coefficient is generated by a collocation point and a boundary element of the same region. Since multiple regions exist, a sparse coefficient matrix $A$ is generated. In QMM method, every dielectric region is decomposed into several fictitious medium blocks. This could increase the sparsity of matrix $A$ greatly. With the storage technique of sparse blocked matrix and iterative equation solver, the computing time and memory usage for the original problem will be greatly reduced. It should be pointed out that the QMM method adds some unknowns to the overall problem, on the additional interfaces between the quasi-multiple mediums. With suitable decomposition, these unknowns would only account for a little percentage of the whole, which would not outweigh the advantage of the increased matrix sparsity. This is well satisfied in the cases of VLSI interconnects, where most boundary elements are located on the conductor surfaces.
Neumann boundaries, where electrical flux is 0.

Figure 1: A typical 3D interconnect capacitor with 5 dielectric layers is cut into 3×2 structure.

In the existing applications of QMM method to finite-domain electrostatic computation, two groups of planes parallel to the YOZ and ZOX planes respectively are set to cut all dielectric layers into pieces (see Fig. 1). Thus, in the top view of the 3D simulated domain, each original dielectric becomes an array of $m \times n$ fictitious medium blocks. We call $(m, n)$ the QMM cutting number. Also, the cutting planes are dispersed along the X-axis or Y-axis uniformly for simplicity. The QMM cutting number influences the efficiency of QMM method directly, and in Section 3.1 a new approach to generate it is given.

2.3 Performance analysis of the QMM acceleration

The CPU time of large-scale electrostatic BEM simulation mainly consists of two parts: $t_{gen}$ and $t_{sol}$. The former is the time spent in generating the coefficient matrix $A$ in (4), while the latter is for solving equation (4). For a multi-region problem, there are the following nearly linear relationships, while using the technology of storing sparse matrix and efficient iterative solver such as GMRES.

$$t_{gen} \propto Z \quad t_{sol} \propto Z \cdot k,$$

where $Z$ stands for the number of non-zero entries of matrix $A$, and $k$ stands for the number of iterations. Because $k$ is much less than the parameter $Z$, the latter become significant to influence the total computing time. Approximately, the fewer non-zero entries there is, the less CPU time will be taken. The reduction of $Z$-value well reflects the computational efficiency of the QMM method [3].

The memory usage also consists of two main parts: $Mem_A$ is used to store the coefficient matrix, and $Mem_v$ is used to store the orthogonal basis vectors in GMRES algorithm. Similarly, the former is proportional to the $Z$-value, while the latter proportional to the product of $Z$-value and the number of unknowns. Using QMM method, $Mem_A$ is reduced by the same ratio with the reduction of $Z$-value, whereas $Mem_v$ is increased because more unknowns are involved. Because $Mem_A$ is usually much larger than $Mem_v$, the total memory usage will reduce
while using QMM method, and the experiments on VLSI interconnects verified
the analysis [2-3]. By the way, a storing scheme of mixed precision is used here
to save memory usage while preserving high computational accuracy, because
only the basis vectors relevant to the orthogonalization process in GMRES are
required to be double precision [9].

3 The enhanced QMM technology

In this section, we firstly present an approach to generate a better QMM cutting
number, and then discuss the issues relevant to the organization and
preconditioned solution of the global linear system.

3.1 The SMZ approach to selecting the QMM cutting number

The approach to select the QMM cutting number is named SMZ (selection with
minimal Z-value), where the numbers of non-zero matrix entries (Z-value) are
calculated in advance, for a number of candidates of \((m, n)\). Eventually, an
optimal cutting number is selected to improve the computational speed of the
QMM-accelerated BEM. According to the analysis in Section 2.3, it is
reasonable to use the Z-value to predict the computational time, and the minimal
Z-value would mean the fastest computational speed. To calculate the Z-values
for different QMM cutting numbers prior to implementing the QMM
decomposition and following computations is the basic idea of the SMZ
approach. Fig. 2 shows the flowchart of the SMZ approach.

Figure 2: The flowchart of the SMZ approach to selecting the cutting number.

In the first step of SMZ approach, a set \(S\) to contain the candidates of QMM
cutting number is determined. Some principles are considered to reduce the
candidates of \((m, n)\) in the set \(S\) for the sake of saving time. Without loss of
generality, the value range of \(m\) (which stands for the number of fictitious
medium blocks along the Y-axis) is discussed. Firstly, \(m\) should not be too small;
otherwise QMM could not result in great acceleration. Secondly, too large value
of \(m\) is also not advisable because the great deal of additional elements on
fictitious interfaces would overwhelm the advantage of QMM cutting, resulting
in the decline of computational efficiency, both in CPU time and memory usage.
So, a moderate value range should be taken for \(m\), according to the domain
geometry. Besides, further limitation for \((m, n)\) could work for the simulated
domain with large aspect-ratio. With above considerations, the number of elements in S becomes relatively small. This makes the additional CPU time for the SMZ algorithm not to influence the whole computational efficiency greatly. At the same time, the S is also big enough to find an optimal cutting number.

Now, how to calculate the Z-value for a giving cutting number is being analyzed. If the region i involves $V_i$ discretized boundary unknowns ($u$ or $q$), there are $Z_i = N_i \cdot V_i$ non-zero coefficients in the $N_i$ discretized BIEs (2), which becomes the non-zero entries of matrix $A$ finally. The $N_i$ is the number of the collocation points in region i, whose quantity equals to the number of boundary elements in region i. The boundary elements are classified into two kinds: the first kind includes those on the Dirichlet or Neumann boundary, where only one unknown of $u$ or $q$ exists on each element; the second kind includes those on region interface, where two unknowns of $u$ and $q$ exist on each element. So,

$$N_i = (a_i + b_i), \quad V_i = (a_i + 2b_i),$$

where $a_i$ stands for the number of first kind boundary elements, $b_i$ stands for the number of second kind boundary elements, both in dielectric region i. Then,

$$Z_i = N_i \cdot V_i = (a_i + b_i)(a_i + 2b_i),$$

Sum up $Z_i$ for all dielectric regions, we get:

$$Z = \sum_{i=1}^{Q} Z_i = \sum_{i=1}^{Q} (a_i + b_i)(a_i + 2b_i),$$

where $Q$ is the total number of dielectric regions. Without QMM cutting, $Q$ equals to the dielectric number $M$; for the computation with the QMM cutting number $(m, n)$, $Q$ equals to $M \times m \times n$. Therefore, the numbers of two kinds of boundary elements in each region are firstly calculated, and then the Z-value is obtained with (8). The calculation of $a_i$ and $b_i$ is related to the element partition method. For a certain partition method, the number of elements on each boundary surface can be calculated with the surface geometry dimension and then be counted in $a_i$ or $b_i$, therefore the $m\times n$ structures need not be actually generated.

After getting the Z-values for all cutting numbers in S, some cutting numbers with nearly minimal Z-value are selected. Among them, the one with the least product of $m\times n$ becomes the optimal cutting number, in order to balance the memory usage and the expected computational speed. Using this optimal QMM cutting number, computational accuracy and memory usage of QMM method are both preserved while achieving higher computational speed.

### 3.2 Organizing the coefficient matrix

Using QMM method, the dielectric regions are at least several times more than the original structure. For example, the structure in Fig. 1 involves 30 regions. If the discretized unknowns were arranged without serious consideration, the population of the coefficient matrix $A$ would be very chaotic. For the problem comprising a large number of regions, a lot of additional CPU time will spent on the switching and locating manipulations among non-zero entries, in each matrix-vector multiplication. It weakens the efficiency of equation solution.
An efficient arrangement of the unknowns and collocation points in constructing the global linear system for multi-region BEM computation was proposed [10]. By this arrangement, the number of non-zero blocks is deceased to the least, and their distribution is so regular that an efficient storage structure can be easily found to save the additional CPU time.

![Figure 3: The distribution of the non-zero matrix entries for (a) a problem with three dielectrics, and (b) a problem including 12 dielectric regions.](image)

With this equation organization, the coefficient matrix $A$ for a three-region problem is shown in Fig. 3(a), where the orders of different unknowns and collocation points related to the dielectric regions are marked aside. Fig. 3(b) shows the non-zero block distribution for a three-dielectric structure applied $2 \times 2$ QMM cutting (totally 12 regions), as another example. This figure shows the regularity of the non-zero block distribution by our approach. It also shows the great matrix sparsity generated by the QMM method.

### 3.3 A new preconditioner for GMRES solution

In the organization of matrix $A$, the order of the collocation points is consistent with that of unknowns, so that the diagonal entries of the matrix are obtained by the singular integrals, which results in a non-zero entry with large absolute value. For this reason, the Jacobi preconditioner can bring quick convergence. Here another easily computed preconditioner is found to give better performance than the Jacobi preconditioner, because of its good approximation to the inverse of $A$.

Similar to the preconditioner presented in [11], the new preconditioner matrix is computed by inverting a sequence of $2 \times 2$ reduced $A$ matrices, one associated with each collocation point. The reduced matrix consists of the $A$'s entries representing the interactions among the current collocation and its one neighbour. Because of the regular arrangement of the matrix $A$, the reduced matrices are mostly the $2 \times 2$ diagonal block of $A$. But for the collocation on interface, its neighbour is the other occurrence of itself, rather than its adjacent entry in $A$.

More than 100 structures of VLSI interconnects are computed to compare the efficiency of new preconditioner and the Jacobi preconditioner. The results show the new method reduces about 30% iteration times than the Jacobi preconditioner,
while the time for equation solution is reduced for 26%, on an average.

4 Numerical Results

More than 100 interconnect structures from actual VLSI layout have been simulated with the enhanced QMM technology. Here we give the numerical results of two examples. The first example is the structure shown in Fig. 1 (The 3×2 QMM cutting is not imposed). There are 18 conductor blocks embedded in five dielectric layers. The simulation region is about 10.7×4.3×5.8 (in μm). The second example is a rather big case with the dimension of 12×109×5.7 (in μm), and there are 34 conductors embedded in five dielectric layers, with one specified as master conductor. The charges on all conductors are computed on a Sun Ultra Enterprise 450 Server with 248 MHz frequency. The stopping criteria of the GMRES solver is set to be $1.0 \times 10^{-4}$. The boundary element partition in the following computations is using a strategy of non-uniform density partitioning to achieve high accuracy with fewer elements [2-3].

![Figure 4: The correlation of computing time of QMM accelerated BEM and the Z-value, for (a) a small structure and (b) a large structure.](image)

Firstly, an experiment is designed to show the correlation between the total computational time of QMM accelerated BEM simulation and the number of non-zero matrix entries involved, i.e. the Z-value. For the first example, 40 different QMM cutting numbers from set $\{(m,n) | 2 \leq m \leq 9, 2 \leq n \leq 6\}$ are manually specified to perform computations. The Z-values and corresponding CPU times for these computations are shown in Fig. 4(a). For the second example, 54 cutting numbers from set $\{(m,n) | 2 \leq m \leq 7, 2 \leq n \leq 10\}$ are used, and the corresponding data are shown in Fig. 4(b). The computing time increases with the Z-value. For the larger structure, the nearly linear relationship between the computing time and the Z-value is demonstrated much better, but even for the smaller first structure it is ideal to use the Z-value to predict the computing time in advance. This experiment verifies the analysis in Section 2.3 and also supports the basic idea of our SMZ approach to selecting QMM cutting number.

Now, our BEM program with the SMZ approach is used to calculate these two
examples. Computational results are shown in Table 1, with comparison to that of the conventional BEM, where we just specify the QMM cutting number to be (1, 1). For the first example, the SMZ approach selects (7, 3) as the optimal cutting number from five candidates. This procedure costs 0.09 second. For the second example, the SMZ approach selects (5, 6) as the optimal cutting number from 15 candidates with 0.48 second. The time consumption of the SMZ approach is no more than 6% of the total computing time. Compared with the BEM without QMM technology, the speed-up ratio is 3.5 and 10.7 respectively, for the two examples. At the same time, the memory usage for BEM with enhanced QMM is much less than that for the conventional BEM.

In Table 1, the charges of the master conductor are listed. To evaluate the accuracy of these computational results, a commercial software Raphael is used to calculate the same structures on the same machine. Raphael utilizes the finite difference method and has a good reputation for computational accuracy in the area of VLSI interconnects, while using high grid density. For the first example, Raphael uses 130.5 second to get a result of $800 \times 10^{18}$ with about 250,000 grids. For the second example, Raphael uses 656 second to get a result of $20217 \times 10^{18}$ with about 1,000,000 grids. Therefore, the discrepancies in the computational results of our QMM accelerated BEM, the conventional BEM, and Raphael are within 3%. However, the speed-up ratio of the BEM with enhanced QMM to Raphael is 77 and 80, respectively.

Table 1. Comparisons between the conventional BEM and QMM accelerated BEM.

<table>
<thead>
<tr>
<th>case</th>
<th>$T_0$(s)</th>
<th>Cut-N</th>
<th>#Ele / #Unk</th>
<th>#Iter</th>
<th>Time(s)</th>
<th>Mem(MB)</th>
<th>Res($10^{18}$C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex1</td>
<td>N/A</td>
<td>(1, 1)</td>
<td>1590 / 1813</td>
<td>17</td>
<td>6.16</td>
<td>4.65</td>
<td>791</td>
</tr>
<tr>
<td>Ex1</td>
<td>0.09</td>
<td>(7, 3)</td>
<td>2142 / 2795</td>
<td>18</td>
<td>1.75</td>
<td>3.22</td>
<td>807</td>
</tr>
<tr>
<td>Ex2</td>
<td>N/A</td>
<td>(1, 1)</td>
<td>5451 / 5976</td>
<td>29</td>
<td>87.7</td>
<td>54.4</td>
<td>19618</td>
</tr>
<tr>
<td>Ex2</td>
<td>0.48</td>
<td>(5, 6)</td>
<td>6376 / 7655</td>
<td>28</td>
<td>8.18</td>
<td>10.5</td>
<td>20211</td>
</tr>
</tbody>
</table>

$T_0$: the CPU time of the SMZ approach to select the cutting number.
Cut-N: QMM cutting number.
#Ele: number of elements (all constant order).
#Unk: number of unknowns.
#Iter: number of iterations by using the new preconditioned GMRES solver.
Time: total CPU time.
Mem: memory usage.
Res: the computational result, the charge on the master conductor.

5 Conclusion

The presented methods enhance the QMM technology in computational speed and adaptability for large finite-domain electrostatic problem of VLSI interconnects. The proposed SMZ approach utilizes the $Z$-value as the predictor of computing time, and automatically selects an optimal QMM cutting number for QMM accelerated BEM. The SMZ approach expands the adaptability of the
QMM technology compared with the approach of specifying QMM cutting number manually or by an empirical formula. The organization of the global system of linear equations and a new GMRES preconditioner are also presented to improve the efficiency of QMM technology. The BEM with enhanced QMM technology was demonstrated to have good computational performance for the practical electrostatic simulation of VLSI interconnects.

Acknowledgement. This work is supported by a China National Foundation for Key Basic Research grant G1998030404, and the National High Technology Development 863 Program of China under grant 2002AA1Z1460SOC.

References


