3D Capacitance Extraction based on Multi-Level Hierarchical Schur Algorithm

Zhifeng Sheng, Patrick Dewilde and Nick van der Meijs
Circuits and Systems Section Department of Micro Electronics and Computer Engineering
Faculty for Electrical Engineering, Mathemathics and Computer Science (EEMCS) Delft University of Technology
Mekelweg 4 NL 2628 CD Delft, The Netherlands
z.sheng,p.m.dewilde,n.p.vandermeijs@tudelft.nl

***Abstract*—In this paper, we study the sparse inverse approximation algorithm used in SPACE, the Layout-to-Circuit Extractor. We briefly introduce SPACE and discuss its limitations. Then, we propose some solutions and demonstrate their efficiency and accuracy with some numeric experiments.**

I. Introduction to SPACE

Parasitic capacitances of interconnects in integrated circuits has become more important as the feature sizes on the circuits are decreased and the area of the circuit is unchanged or increased. For sub-micron integrated circuits - where the vertical dimensions of the wires are in the same order of magnitude as their minimum horizontal dimensions - 3D numerical techniques are even required to accurately compute the values of the interconnect capacitances.

SPACE is a layout-to-circuit extraction program, that is used to accurately and efficiently compute 3D interconnect capacitances of integrated circuits based upon their mask layout description. The 3D capacitances are part of an output circuit together with other circuit components like transistors and resistances. This circuit can directly be used as input for a circuit simulator like SPICE.

The boundary element method that is adapted in SPACE can be described briefly as follows:

1) For the purpose of modeling IC interconnections, it is sufficient to suppose that the chip is stratified medium in which the conductors are floating. For such a medium, the potential at a point p can be written as:

$$\Phi(p) = \int_{\mathcal{D}} G(p,q)\rho(q)\mathrm{d}q, q \in \mathcal{D} \tag{1}$$

where the Green's function $G(p,q)$ represents the potential induced at point p, due to a unit point charge at point q. In this thesis, the Green's function $G(p,q)$ is computed with the single integration formula presented in [1].

2) The above equations are transformed into a matrix equation by discretizing the surface charge on the conductors as a piecewise linear and continuous distribution on a set of boundary elements.
3) Let N be the total number of boundary elements, the matrix equation can be written as:

$$\Phi = \mathbf{G}\sigma \tag{2}$$

where $\Phi = [\phi_1, \phi_2 ... \phi_N]^T$ and $\sigma = [\sigma_1, \sigma_2 ... \sigma_N]^T$ collect the potentials of the boundary elements and charges on the boundary elements respectively, and $G_{ij}\sigma_j$ is the potential induced at the element i by the charge at the boundary element j.

4) Using this equation, we can compute the conductor capacitance as follows: Let $\mathbf{A}$ be an incidence matrix relating each boundary elements to the conductors, i.e. $\mathbf{A}_{ij} = 1$ if element i lies on conductor j, and $\mathbf{A}_{ij} = 0$ if otherwise. Also, let M be the total number of conductors, $\mathbf{V} = [V_1, V_2, ..., V_M]^T$ be the vector of conductor potentials and $\mathbf{Q} = [Q_1, Q_2, ..., Q_M]^T$ be the vector charges on the conductors, then:

$$\mathbf{Q} = \mathbf{A}^T\mathbf{G}^{-1}A\mathbf{V} = \mathbf{C}_s\mathbf{V} \tag{3}$$

Hence:

$$\mathbf{C}_s = \mathbf{A}^T\mathbf{G}^{-1}\mathbf{A}\mathbf{V} \tag{4}$$

5) The matrix $\mathbf{C}_s$ is the short circuit capacitance matrix. The capacitance network is derived from the short circuit capacitance matrix as follows:

$$C_{ij} = -C_{sij} \text{ for } i \neq j, C_{ii} = \sum_{j=1}^{M} C_{sij} \tag{5}$$

Consequently, the matrix $\mathbf{G}$ has to be generated and inverted. This matrix can be very big and full. Generating and inverting such a matrix is prohibitively expensive. Moreover, the full matrix would result in a too complicate circuit for sensible verification.

As a solution, SPACE adapts a scan-line algorithm, the generalized Schur algorithm and the hierarchically Schur algorithm to compute a sparse inverse approximation of $\mathbf{G}^{-1}$. Thereby in effect ignoring small capacitances between conductors that are physically "far" from each other. Let w be the parameter denotes the distance over which capacitive coupling is significant. The CPU time and memory complexity of SPACE are $O(Nw^4)$ and $O(w^4)$ respectively, where N is the total number of boundary elements, and the parameter w denotes the distance over which capacitive coupling is considered to be significant.

For more details about the boundary element analysis, scan-line algorithm, the generalized Schur algorithm and the

hierarchically Schur algorithm, please refer to the PhD thesis of N. P. van der Meijs [2].

II. Limitations of the algorithms used in SPACE

Although, SPACE is very efficient in generating the capacitance network for 3D layouts, we believe the underlying algorithms do have some limitations and we can improve them easily.

1) Although SPACE extracts capacitance networks for three dimensional layouts, we may describe its algorithm as 2.5D in the sense that it assumes the vertical dimension of the layouts to be very thin. This assumption is quite valid at the time when SPACE came out. However, after many years of development in VLSI technology, circuits with many more layers are common and the vertical dimension can not be ignored anymore. In fact, if we assume the vertical dimension to be comparable with the horizontal dimensions, the CPU time complexity of SPACE quickly becomes $O(N^{5/3}w^4)$ which is not linear in the total number of panels.
2) Again, when the vertical dimension is not ignorable, the memory complexity of SPACE becomes $N^{2/3}w^4$, which not only means that much more memory is needed, but also indicates that much more entities in the Green's function matrix must be computed. And the computation of Green's functions is a major factor of the CPU time needed.
3) Due to historical reasons, SPACE adapt the Hierarchical Schur algorithm in the X axis and then apply the Schur algorithm on the Y axis. This is not a very consistent scheme, in the sense that, with this kind of scheme, SPACE with exactly the same configuration would generate different capacitance network for the same layout depending on which direction the layout aligns with.

Out of the considerations above, we propose a multi-level hierarchical Schur algorithm which we shall present in the following sections.

III. Multi-Level Hierarchical Schur Algorithm

The straight-forward idea is to apply the hierarchical Schur algorithm along both the X and Y axis. And we can even go further by applying the hierarchical Schur algorithm along X, Y and Z axises. In this way, we can efficiently deal with a genuine three dimensional layout. In this section, we refer to these algorithms as multi-level hierarchical Schur algorithms.

A. Notations

Before, we present the Multi-level hierarchical Schur algorithms, we would like to introduce a few notations that will be used consistently hereafter in this chapter.

- Let the 3D layout of interconnects be discretized with boundary elements; L_x, L_y, L_z be its maximum dimensions in x, y, z axises, respectively.
- Let x_0, y_0, z_0 be the smallest coordinates of the layout in x, y and z axises, we have the layout completely embedded in the bounding box $\{\Omega; x_0 \leq x \leq x_0 + L_x, y_0 \leq y \leq y_0 + L_y, z_0 \leq z \leq z_0 + L_z\}$.
- Assume a certain 2D scan-window of dimensions $w_x \times w_y$, $\{\Omega(i,j); 1 \leq i \leq \lceil L_x/w_x \rceil, 1 \leq i \leq \lceil L_y/w_y \rceil\}$ denotes a sub-domain that $\{x_0 + (i-1) \times w_x \leq x \leq x_0 + i \times w_x, y_0 + (j-1) \times w_y \leq y \leq y_0 + j \times w_y, z_0 \leq z \leq z_0 + L_z\}$ which is bounded in the 2D scan-window $\mathcal{W}(i,j)$.
- Assume a certain 3D scan-window of dimensions $w_x \times w_y \times w_z$, $\{\Omega(i,j,k); 1 \leq i \leq \lceil L_x/w_x \rceil, 1 \leq i \leq \lceil L_y/w_y \rceil, 1 \leq i \leq \lceil L_z/w_z \rceil\}$ denotes a sub-domain $\{x_0 + (i-1) \times w_x \leq x \leq x_0 + i \times w_x, y_0 + (j-1) \times w_y \leq y \leq y_0 + j \times w_y, z_0 + (k-1) \times w_z \leq z \leq z_0 + i \times w_z\}$ which is bounded in the 3D scan-window $\mathcal{W}(i,j,k)$.
- Let $\cup$ donates a binary merging operator that collects boundary elements from both sub-domains and number them locally, for instance, $\Omega(i,j) \cup \Omega(i,j+1)$ or equivalently $\bigcup_{m=j}^{j+1} \Omega(i,m)$.
- Let $\mathbf{G}$ be an operator which generates a matrix of Green's functions for all boundary elements in a certain domain/sub-domain, for instance, $\mathbf{G}(\Omega(i,j,k))$ produces the matrix $\mathbf{G}_{\Omega(i,j,k)}$ which contains all Green's functions for the boundary elements within the sub-domain $\Omega(i,j,k)$. For this matrix, a local numbering of the boundary elements is used.
- Let $\square\left[\mathbf{G}_{\Omega(i,j,k)}\right]$ or equivalently $\square\left[\mathbf{G}(\Omega(i,j,k))\right]$ denotes a embedding process that takes the matrix $\mathbf{G}_{\Omega(i,j,k)}$ with local numbering and embed it into a larger empty matrix (matrix with only zero entries) according to the map between the local numbering and global numbering of the boundary elements. The embedding operator can be specified more precisely. Assume there are M boundary elements that are locally numbered in $\Omega(i,j,k)$, and there are N boundary elements that are globally numbered in Ω. Then there is a unique incidence matrix $\mathbf{I}_{\Omega(i,j,k)}$ of dimension $M \times N$ that maps the local indexes to the global indexes, i.e $\mathbf{I}_{\Omega(i,j,k)}(m,n) = 1$ if the boundary with local index m is numbered with global index n. $\mathbf{I}_{\Omega(i,j,k)}(m,n) = 0$, if otherwise. The transpose of this incidence matrix maps the global indexes to the local indexes. And we have:

$$\square\left[\mathbf{G}_{\Omega(i,j,k)}\right] = \mathbf{I}_{\Omega(i,j,k)} \mathbf{G}_{\Omega(i,j,k)} \mathbf{I}^T_{\Omega(i,j,k)}. \tag{6}$$

  And apparently, the incidence matrix has the following property:

$$\mathbf{I}^T_{\Omega(i,j,k)} \mathbf{I}_{\Omega(i,j,k)} = \mathbf{I} \tag{7}$$

  Here, $\mathbf{I}$ denotes a identity matrix of dimensions $M \times M$.
- $\mathbf{G}^{-1}_{\mathrm{SI}}$ denotes a sparse approximation of $\mathbf{G}^{-1}$.

B. Two dimensional scan-window algorithm

With the whole layout discretized with boundary elements and then segmented with 2D scan-windows of size $w \times w$, the

sparse inverse to the $\mathbf{G}$ is defined as:

$$\begin{aligned}\mathbf{G}_{\mathrm{SI}}^{-1} = \sum_{i=1}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\square[\mathbf{G}^{-1}(\bigcup_{m=i}^{i+1}\bigcup_{n=j}^{j+1}\Omega(m,n))] \\ -\sum_{j=2}^{\lceil L_y/w\rceil-1}\square[\mathbf{G}^{-1}(\bigcup_{m=i}^{i+1}\Omega(m,j))]\} \\ -\sum_{i=2}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\square[\mathbf{G}^{-1}(\bigcup_{n=j}^{j+1}\Omega(i,n))] \\ -\sum_{j=2}^{\lceil L_y/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega(i,j))]\} \quad (8)\end{aligned}$$

Then we may replace the $\mathbf{G}^{-1}$ in Eq. (4) with the above sparse approximation and we have:

$$\begin{aligned}\mathbf{C}_s \approx \mathbf{A}^T\sum_{i=1}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\square[\mathbf{G}^{-1}(\bigcup_{m=i}^{i+1}\bigcup_{n=j}^{j+1}\Omega(m,n))] \\ -\sum_{j=2}^{\lceil L_y/w\rceil-1}\square[\mathbf{G}^{-1}(\bigcup_{m=i}^{i+1}\Omega(m,j))]\} \\ -\sum_{i=2}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\square[\mathbf{G}^{-1}(\bigcup_{n=j}^{j+1}\Omega(i,n))] \\ -\sum_{j=2}^{\lceil L_y/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega(i,j))]\}\mathbf{A} \quad (9)\end{aligned}$$

Let $\mathbf{C}_s(\Omega(i,j)) = \mathbf{A}^T\square[\mathbf{G}^{-1}(\Omega(i,j))]\mathbf{A}$ denote the (partial) short-circuit capacitance matrix generated for $\Omega(i,j)$. With the definition of the embedding operator, We can write

$$\mathbf{C}_s(\Omega(i,j)) = \mathbf{A}^T\mathbf{I}_{\Omega(i,j)}\mathbf{G}^{-1}(\Omega(i,j))\mathbf{I}_{\Omega(i,j)}^T\mathbf{A} \quad (10)$$

We also have a local incidence matrix $\mathbf{A}_{\Omega(i,j)}$ that relates boundary elements with local indexes to conductor potential, i.e $\mathbf{A}_{\Omega(i,j)}(m,n) = 1$ if the boundary element with the local index m lies on the conductor n, and 0 if otherwise. Due to the fact that Eq. (10) only involves the boundary elements in $\Omega(i,j)$, the additional information in $\mathbf{A}$ that counts other boundary elements will not be taken into account. Therefore, we may compute the global approximated short-circuit capacitance matrix as:

$$\begin{aligned}\mathbf{C}_s \approx \sum_{i=1}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\mathbf{C}_s(\bigcup_{m=i}^{i+1}\bigcup_{n=j}^{j+1}\Omega(m,n)) \\ -\sum_{j=2}^{\lceil L_y/w\rceil-1}\mathbf{C}_s(\bigcup_{m=i}^{i+1}\Omega(m,j))\} \\ -\sum_{i=2}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\mathbf{C}_s(\bigcup_{n=j}^{j+1}\Omega(i,n)) - \sum_{j=2}^{\lceil L_y/w\rceil-1}\mathbf{C}_s(\Omega(i,j))\} \quad (11)\end{aligned}$$

Explicit computation of the matrix inverse is not recommended. Therefore, we may compute $\mathbf{G}^{-1}(\Omega(i,j))\mathbf{A}_{\Omega(i,j)}$ by solving system of linear equations, $\mathbf{G}(\Omega(i,j))\mathbf{x} = \mathbf{A}_{\Omega(i,j)}$. Both direct solution method and iterative solution method can be used here. Note that, global indexes and global incidence matrices do not appear in the above formula and we do not have to compute them explicitly. All computations can remain local. As soon as a (partial) short-circuit capacitance matrix is generated, we will use it to modified the short-circuit capacitance matrix of the whole layout. Therefore, at any instant, the program only has to analyse a (small) segment of the whole circuit. This enables the algorithm to deal with large circuit with consuming little computer memory. For more details, we refer to the PhD thesis of Dr. N. P. van der Meijs [2] and we invite interested readers to look into prototype that we have implemented. Let the lengths in the three axises be comparable, the total number of two dimensional scan-windows is of $O(N^{2/3}w^{-2})$. The total number of panels inside each scan-window is of order $O(N^{1/3}w^2)$. A system of linear equations are to be solved in each scan-window and the complexity is of $O(w^6N)$. Therefore, assuming the lengths in the three axises be comparable, the CPU time complexity of this algorithm is of $O(N^{5/3}w^4)$. and its memory complexity is of $O(N^{2/3}w^4)$.

C. Three dimensional scan-window algorithm

Similarly, we may apply the scan-window algorithm along X, Y and Z axises. And we assume the whole layout be discretized with boundary elements and then segmented with 3D scan-windows of size $w \times w \times w$, the sparse inverse to the $\mathbf{G}$ is then defined as:

$$\begin{aligned}\mathbf{G}_{\mathrm{SI}}^{-1} = \sum_{i=1}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\{\sum_{k=1}^{\lceil L_z/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega_1)] \\ -\sum_{k=2}^{\lceil L_z/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega_2)]\} - \sum_{j=2}^{\lceil L_y/w\rceil-1}\{\sum_{k=1}^{\lceil L_z/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega_3)] \\ -\sum_{k=2}^{\lceil L_z/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega_4)]\}\} \\ -\sum_{i=2}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\{\sum_{k=1}^{\lceil L_z/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega_5)] \\ -\sum_{k=2}^{\lceil L_z/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega_6)]\} \\ -\sum_{j=2}^{\lceil L_y/w\rceil-1}\{\sum_{k=1}^{\lceil L_z/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega_7)] - \sum_{k=2}^{\lceil L_z/w\rceil-1}\square[\mathbf{G}^{-1}(\Omega_8)]\}\} \quad (12)\end{aligned}$$

where:

$$\begin{aligned}\Omega_1 &= \bigcup_{l=i}^{i+1}\bigcup_{m=j}^{j+1}\bigcup_{n=k}^{k+1}\Omega(l,m,n), \quad \Omega_2 = \bigcup_{l=i}^{i+1}\bigcup_{m=j}^{j+1}\Omega(l,m,k), \\ \Omega_3 &= \bigcup_{l=i}^{i+1}\bigcup_{n=k}^{k+1}\Omega(l,j,n), \quad \Omega_4 = \bigcup_{l=i}^{i+1}\Omega(l,j,k), \\ \Omega_5 &= \bigcup_{m=j}^{j+1}\bigcup_{n=k}^{k+1}\Omega(i,m,n), \quad \Omega_6 = \bigcup_{m=j}^{j+1}\Omega(i,m,k), \\ \Omega_7 &= \bigcup_{n=k}^{k+1}\Omega(i,j,n), \quad \Omega_8 = \Omega(i,j,k). \quad (13)\end{aligned}$$

Let $\mathbf{C}_s(\Omega_l) = \mathbf{A}^T(\Omega_l)\mathbf{G}^{-1}(\Omega_l)\mathbf{A}(\Omega_l)$ denote the (partial) short-circuit capacitance matrix generated for $\Omega_l; l \in \{1,2,3,4,5,6,7,8\}$, we may compute the approximated short-circuit capacitance matrix as:

$$\begin{aligned}\mathbf{C}_s \approx \sum_{i=1}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\{\sum_{k=1}^{\lceil L_z/w\rceil-1}\mathbf{C}_s(\Omega_1) - \sum_{k=2}^{\lceil L_z/w\rceil-1}\mathbf{C}_s(\Omega_2)\} \\ - \sum_{j=2}^{\lceil L_y/w\rceil-1}\{\sum_{k=1}^{\lceil L_z/w\rceil-1}\mathbf{C}_s(\Omega_3) - \sum_{k=2}^{\lceil L_z/w\rceil-1}\mathbf{C}_s(\Omega_4)\}\} \\ - \sum_{i=2}^{\lceil L_x/w\rceil-1}\{\sum_{j=1}^{\lceil L_y/w\rceil-1}\{\sum_{k=1}^{\lceil L_z/w\rceil-1}\mathbf{C}_s(\Omega_5) - \sum_{k=2}^{\lceil L_z/w\rceil-1}\mathbf{C}_s(\Omega_6)\} \\ - \sum_{j=2}^{\lceil L_y/w\rceil-1}\{\sum_{k=1}^{\lceil L_z/w\rceil-1}\mathbf{C}_s(\Omega_7) - \sum_{k=2}^{\lceil L_z/w\rceil-1}\mathbf{C}_s(\Omega_8)\}\}\end{aligned}$$

Again, explicit computation of the matrix inverse is not recommended. Therefore, we may compute $\mathbf{G}^{-1}(\Omega_l)\mathbf{A}(\Omega_l)$ by solving system of linear equations, $\mathbf{G}(\Omega_l)\mathbf{x} = \mathbf{A}(\Omega_l)$. Both direct solution method and iterative solution method can be used here.

Note that, as soon as a (partial) short-circuit capacitance matrix is generated, we will use it to modified the short-circuit capacitance matrix of the whole layout. Therefore, at any instant, the program only has to analyse a (small) segment of the whole circuit. This enables the algorithm to deal with large circuit with consuming little computer memory.

Let the lengths in the three axises be comparable, the number of three dimensional scan-window is of $O(Nw^{-3})$. The total number of panels inside each scan-window is of order $O(w^3)$. A system of linear equations are to be solved in each scan-window and the complexity is of $O(w^9)$. Therefore, assuming the lengths in the three axises be comparable, the CPU time complexity of this algorithm is of $O(Nw^6)$. and its memory complexity is of $O(w^6)$.

D. Numeric result

To test the accuracy and complexity of these algorithms above, we implemented a random layout generator which grows random conductors in a three dimensional domain. Through the random layout generator, one can specify the boundary of the three dimensional domain, the number of conductors to be generated and the maximum length of each conductor. These randomly generated conductors do not overlap with each other and each conductor is simply connected. In Fig 1, we show a few examples of the layouts generated.

To study the accuracy and computational cost of these algorithms, we generate a layout consisting of 20 conductors each with 100 unit of length as shown in Fig. 1(b), and then compute its short-circuit capacitance matrices with one dimensional scan-line algorithm, two dimensional scan-window algorithm and three dimensional scan-window algorithm, respectively. The results are compared with the exact solution and the relative mean square errors defined in Eq. (14) are computed.

$$\text{RMSE}_c = \frac{\|\mathbf{C}_s^{\text{approx}} - \mathbf{C}_s^{\text{exact}}\|}{\|\mathbf{C}_s^{\text{exact}}\|} \tag{14}$$

The layout as shown in Fig. 1(b) is computed with different algorithms combined with scan-windows of different sizes. The experimental results are shown in Fig. 2 and Fig. 3.

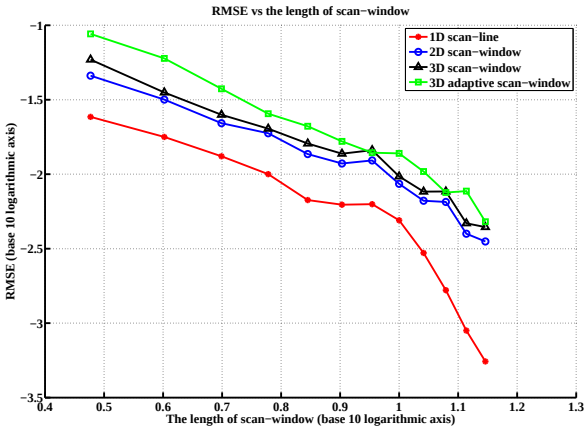


Fig. 2. The relative mean square errors in the computed short-circuit capacitance matrices.

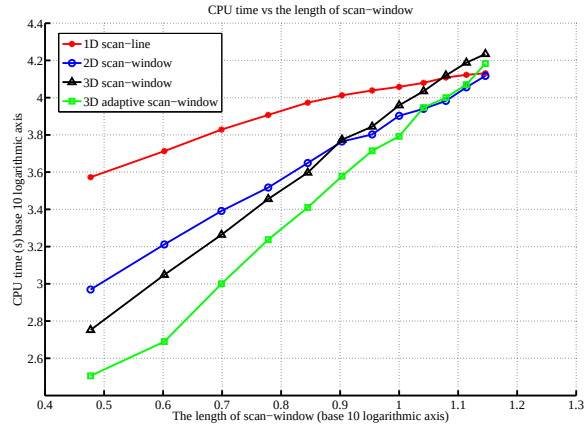
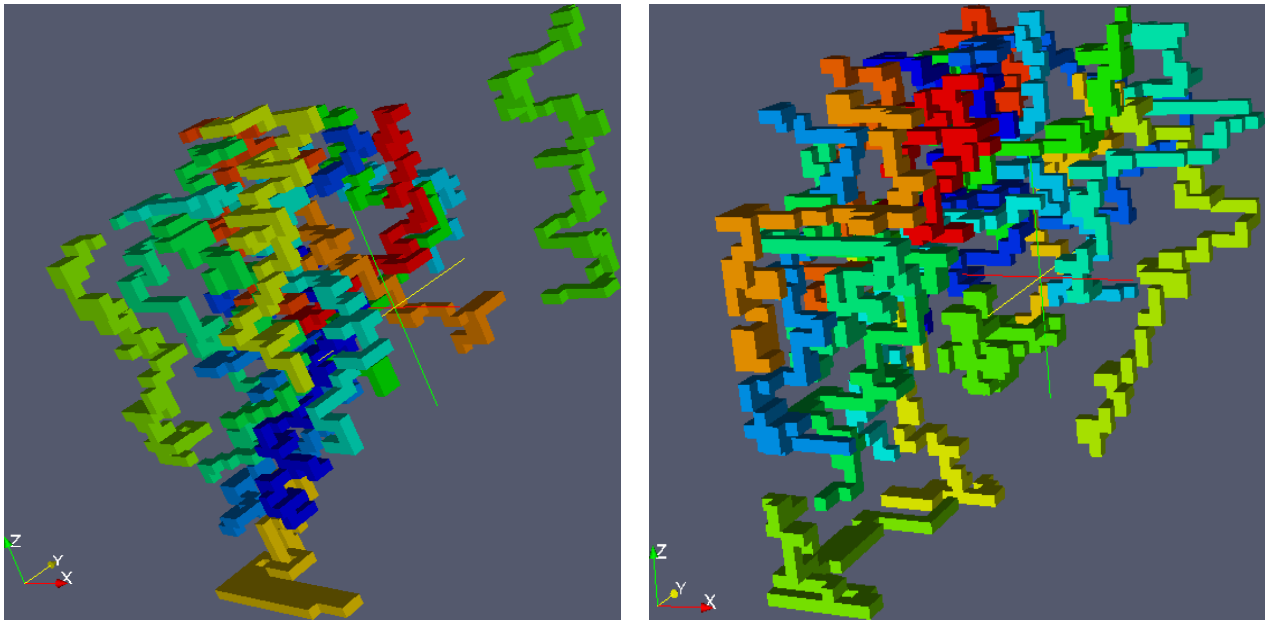


Fig. 3. The CPU time needed to computed the short-circuit capacitance matrices Vs the scan-window size.

Fig. 2 shows that all these scan algorithms are convergent with the increasing window size, and no surprise that with the same size for scan window, 1D scan-line algorithm is more accurate than the 2D scan-window algorithm which is slightly more accurate than the 3D scan-window algorithm. It is also clear that the total RMSE error is dominated by the approximation made at the higher levels. Fig. 2 shows that 3D scan-window algorithm is more efficient than the other two algorithms. It also confirms that the CPU time needed for the 3D scan-window algorithm increases more sharply with the size of the scan-window. Other than these, we also observed that all three



(a) 15 conductors in 40x40x40 (b) 20 conductors in 40x40x40

Fig. 1. The randomly generated layout of conductors in three dimensional domain. The surface mesh of the layout (b) consists of 7172 boundary elements.

algorithms deliver more accurate result when more conductors are clustered together. This because nearby conductors shield each other and local interactions become more dominate.

E. Adaptive three dimensional scan-window algorithm

As shown in Fig. 2, the error made by scanning along the X axis dominates the total relative mean square error. And a larger scan window along Y and Z axis would not help too much as long as the scan-window along X axis is small. Therefore, it is reasonable to use a bigger scan window along the X axis and decrease the size of scan-window along Y axis and Z axis. That is $w_x > w_y > w_z > 0$. The adaptive algorithm is very similar to the original three dimensional scan-window algorithm algorithm, except scan-windows of different sizes are used along X, Y and Z axises.

To study the accuracy and computational cost of the adaptive 3D scan-window algorithm, we compute the short-circuit capacitance matrices of the layout as shown in Fig. 1(b). And then we compare the result with the exact solution. The relative mean square error is defined in Eq. (14).

As shown in Fig. 2 and Fig. 3, the adaptive 3D scan-window algorithm achieves comparable accuracy with much less computational time. Note that, one can apply different schemes to decrease the size of scan-windows, and they may deliver different results.

IV. Summary

In this paper, we proposed a series of efficient scan-window algorithms that can be used in SPACE for capacitance extraction. Numeric experiments have confirmed that the Hierarchical (adaptive) 3D scan-window algorithm is efficient and sufficiently accurate. Due the simplicity of these algorithms, it should be simple to adapt them in SPACE. This would enhance the capacity of SPACE in handling 3D layout of circuits.

References

[1] D. Wilton, S. Rao, A. Glisson, D. Schaubert, O. Al-Bundak, and C. Butler, "Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains," *Antennas and Propagation, IEEE Transactions on*, vol. 32, no. 3, pp. 276–281, Mar 1984.

[2] N. P. van der Meijs, "Accurate and efficient layout extraction," Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands, January, 1992.