RWCap2: Advanced Floating Random Walk Solver for the Capacitance Extraction of VLSI Interconnects

[**Invited Special Session Paper**][1]

Wenjian Yu

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
Email: yu-wj@tsinghua.edu.cn

Abstract

In this paper, we present the techniques in RWCap2, which is a fast capacitance field solver for VLSI interconnects. The solver is based on the floating random walk (FRW) algorithm for capacitance extraction. Hence, it has the advantages of less memory usage, better parallelism and tunable accuracy over the conventional methods for capacitance field solver. To improve the computational speed of the FRW algorithm, and equip it for the extraction task with chip-scale large structures, several techniques are proposed and integrated into RWCap2. We will introduce the usage of the solver and demonstrate its efficiency for capacitance extraction.

1. Introduction

The floating random walk (FRW) algorithm [1-6] is a major field-solver method for capacitance extraction. Compared with the deterministic algorithms (e.g. boundary element method [7, 8]), the FRW algorithm has several advantages, especially for the chip-scale large interconnect structures.

Today, the FRW algorithm has become the kernel of commercial capacitance solvers (such as QuickCap). With parallel computing techniques, they have been applied to the block- or chip-level extraction tasks in the sign-off verification of very large-scale integrated (VLSI) circuits. However, approximations are made in the commercial FRW solvers; the theory and technique of accurate FRW solver for the problem with multi-dielectric environment are not established. Although techniques were recently proposed to deal with arbitrary dielectric configuration and for a fabric-aware extraction problem [9, 10], they are either inefficient for large problem or not for the general problem of capacitance extraction. As for handling the multiple dielectrics in actual applications, the existing works [3, 4] are not efficient or lack technical details.

To fill in the gap between the theory and application of the FRW based capacitance solver, a program set called RWCap [11] was developed in 2012, and upgraded to RWCap2 (Version 2) in May 2013. Efficient techniques were also proposed to parallelize the FRW based capacitance extraction on GPUs [6]. In this paper, the techniques used in RWCap2 are introduced. Due to the page limit, we will emphasize their basic ideas, and the usage of RWCap2, which is available to public.

2. Background

The electric potential of a point $\boldsymbol{r}$ can be expressed as an integral of the potential on a surface S surrounding it:

$$\phi(\boldsymbol{r}) = \oint_S P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \phi(\boldsymbol{r}^{(1)}) \mathrm{d}\boldsymbol{r}^{(1)}, \quad (1)$$

where $\phi(\boldsymbol{r})$ is the potential of point $\boldsymbol{r}$ and $P(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ is called the surface Green's function. The domain enclosed by S is often called the transition domain. $P(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ is non-negative for any point $\boldsymbol{r}^{(1)}$ on S, and can be regarded as the probability density function (PDF) for selecting a random point on S. With the principle of Monte Carlo (MC) simulation, $\phi(\boldsymbol{r})$ can be estimated as the statistical mean of $\phi(\boldsymbol{r}^{(1)})$.

To compute the capacitances related with conductor i (called master conductor), a Gaussian surface G_i is firstly constructed to enclose it (see Fig. 1). According to the Gaussian theorem, charge Q_i of conductor i becomes

$$Q_i = \oint_{G_i} F(\boldsymbol{r}) g \oint_{S^{(1)}} \omega(\boldsymbol{r}, \boldsymbol{r}^{(1)}) P^{(1)}(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \phi(\boldsymbol{r}^{(1)}) d\boldsymbol{r}^{(1)} d\boldsymbol{r} \quad . (2)$$

where $F(\boldsymbol{r})$ is the dielectric permittivity at point $\boldsymbol{r}$, g is a constant, and function $\omega(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ is called weight value [1, 5]. Here $P^{(1)}$ denotes the surface Green's function for $S^{(1)}$ enclosing $\boldsymbol{r}$. With the MC method, Q_i can be estimated as the statistical mean of sampled values on G_i, which is further the mean of sampled values on $S^{(1)}$ multiplying the weight value. When $\phi(\boldsymbol{r}^{(1)})$ is unknown,

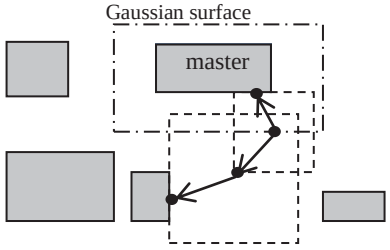


Fig. 1 Two examples of the random walk in the FRW algorithm for capacitance extraction (a 2-D cross-section view).

[1] This paper is invited by Special Session of "Advances in Parasitic Extraction and Circuit Simulation" in ASICON 2013.

The work was supported in part by Beijing Natural Science Foundation (No. 4132047), National Natural Science Foundation of China (No. 61076034), the Opening Foundation of ASIC and System State Key Laboratory (No. 12KF009), and the Tsinghua University Initiative Scientific Research Program.

978-1-4673-6417-1/13/$31.00 ©2013 IEEE

(1) can be applied recursively to (2), which means the sampling procedure repeats until the potential of a sample point is known. This can be interpreted as a floating random walk (FRW) procedure: for the j-th hop of a walk, a transition domain centered at $\boldsymbol{r}^{(j-1)}$ is constructed and then a point $\boldsymbol{r}^{(j)}$ is randomly selected on its boundary according to the discrete probabilities obtained with $P(\boldsymbol{r}^{(j-1)}, \boldsymbol{r}^{(j)})$. The walk terminates after k hops if $\phi(\boldsymbol{r}^{(k)})$ is known, e.g. it is on the surface of a conductor with known potential (see Fig. 1).

Although the surface Green's function for a spherical transition domain has simple analytical expression [2], we only consider the cubic transition domain that is well suited to the Manhattan-shaped interconnects in VLSI circuit. The surface Green's function $P^{(1)}$ in (2) only depends on the relative position of $\boldsymbol{r}^{(1)}$. So, we can pre-calculate and tabulate the sampling probability and weigh value for a unit-size cube.

In a multi-conductor system, the charge of a conductor is a linear combination of the potentials of all conductors. By definition, the capacitances between the conductor and other conductors are the combination coefficients. Therefore, the aforementioned deduction of FRW procedure reveals that the statistical mean of the weight values for the walks terminating at conductor j approximates capacitance C_{ij} between conductors i and j (if $j \neq i$), or the self-capacitance C_{ii} of master conductor i.

The total runtime of the FRW algorithm is roughly:

$$T_{total} = N_{walk} \cdot N_{hop} \cdot T_{hop} \quad , \qquad (3)$$

where N_{walk} is the number of random walks/paths, N_{hop} is the average number of hops in a walk, and T_{hop} is the average computing time for a hop.

3. Advanced Techniques in RWCap2

The first two techniques are able to reduce N_{hop} and N_{walk}, respectively. The last is helpful for reducing T_{hop}.

3.1 Efficient treatment of the multi-layered dielectrics

The analytical surface Green's function for the spherical transition domain with different hemispheres helps the FRW algorithm to handle the situation with multiple dielectrics [2]. Since the original FRW [1] relies on that the transition cube is within a single dielectric, the sphere transition domain can be used to continue the walk stopping at dielectric interface. Fig. 2(a) illustrates this strategy. However, this would induce many hops in a walk, and thus is not efficient.

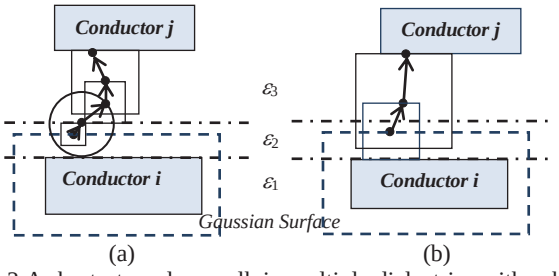


Fig. 2 A shortest random walk in multiple dielectrics with spherical transition domain (a) and precharacterization technique (b).

With numerical techniques like the finite difference method (FDM), the transition probabilities can be calculated for a transition cube with multiple dielectrics [9]. For a cubic domain with multiple dielectrics, as the one in Fig. 3, the electric potential in a homogeneous region is governed by the Laplace equation. Supposing Dirichlet boundary condition (i.e. known potential on the cube surface), we can solve the potential of inside points. This involves the discretization of the 3-D domain or 2-D boundary. And, the solved potential of cube's center point is a linear combination of the potentials of the cube's boundary panels. The combination coefficients are just the transition probabilities that we want.

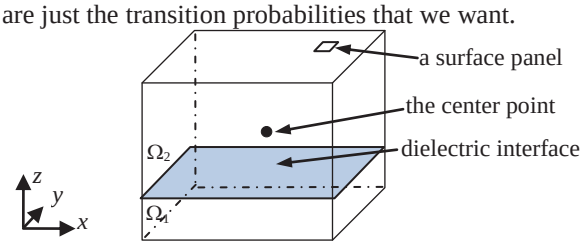


Fig. 3 A transition cube with multiple dielectrics.

Our approach is based on two observations: 1) modern VLSI process technology mainly involves multiple layers of metal wires and dielectrics; 2) the capacitance coupling mainly exists between neighbor layers, and enabling a hop across a dielectric interface would shorten the length of most walks. So, we propose to numerically characterize the surface Green's function for the cubic transition domain with two dielectric layers (see Fig. 3), which would greatly improve the efficiency with limited memory overhead. Fig. 2(b) illustrates the situation where a hop is able to cross one dielectric interface. Comparing Fig. 2(b) and Fig. 2(a), we can see that the number of hops is largely reduced.

Efficient FDM techniques [5] have been developed to solve the Dirichlet boundary-value problem for the transition cube with two dielectric layers and produce the transition probabilities tables (called GFTs) and the weight value tables (called WVTs). The corresponding program is "TechGFT" in the RWCap program set [11].

3.2 Variance reduction for fast MC convergence

In the FRW algorithm, the weight value:

$$\omega(\boldsymbol{r}, \boldsymbol{r}^{(1)}) = -\frac{\nabla_r P^{(1)}(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})}{g P^{(1)}(\boldsymbol{r}, \boldsymbol{r}^{(1)})} = -\frac{\nabla_r P(\boldsymbol{r}', \boldsymbol{r}'^{(1)}) \bullet \hat{n}(\boldsymbol{r}')}{g \cdot L \cdot P(\boldsymbol{r}', \boldsymbol{r}'^{(1)})} \quad (4)$$

is the estimate of capacitance in the MC procedure. In (4), L is the edge length of the first transition cube, $P(\boldsymbol{r}', \boldsymbol{r}'^{(1)})$ denotes the surface Green's function for the unit-size cube, and superscript ' labels the positions in the unit-size cube. Due to the central limit theorem, the capacitance result obeys the normal probability distribution, and its standard deviation (Std) is estimated with the variance of the returned weight values of walks

(zero for the walk not hitting relevant conductor). To improve the convergence rate of the MC procedure, it is crucial to reduce the variance of weight values for a given number of walks. Note that with the help of pre-calculated WVTs, we actually know the possible values of the weight value. In Fig. 4(a), we draw the distributions of the opposite of weight value (4) for the $\boldsymbol{r}^{(1)}$ on top and side faces of the cube. It demonstrates that the weight value has certain degree of variance.

Inspired by importance sampling (IS), we modify the integral formula (2) so as to reduce the variance of resulting new weight value. For the transition cube with a specific dielectric configuration, we first calculate:

$$K = \oint_S \left|\nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})\right| d\boldsymbol{r}^{(1)} \quad , \tag{5}$$

where S is the surface of the unit-size cube. Since $\boldsymbol{r}$ is the center of cube, K is a positive constant. Then,

$$Q_i = \oint_{G_i} F(\boldsymbol{r}) g \oint_S -\frac{\nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})}{L \cdot g} \phi(\boldsymbol{r}^{(1)}) d\boldsymbol{r}^{(1)} d\boldsymbol{r}$$

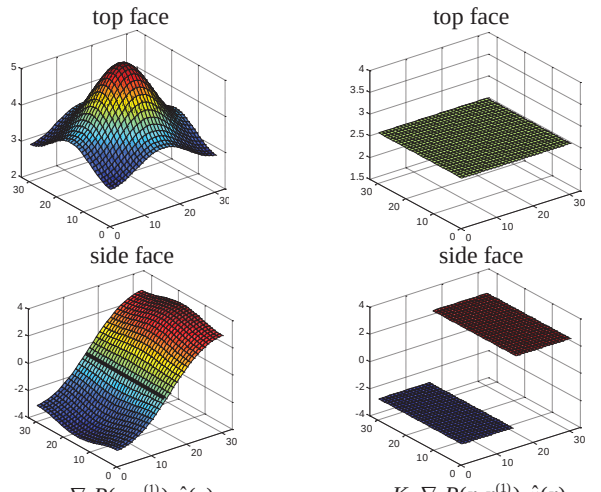
$$= \oint_{G_i} F(\boldsymbol{r}) g \oint_S \frac{-K \cdot \nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})}{L \cdot g \left|\nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})\right|} \cdot \frac{\left|\nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})\right|}{K} \phi(\boldsymbol{r}^{(1)}) d\boldsymbol{r}^{(1)} d\boldsymbol{r}$$

We define function $q(\boldsymbol{r}, \boldsymbol{r}^{(1)}) = \left|\nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})\right| / K$ to be a new PDF for sampling on S. The new weight value is:

$$\tilde{\omega}(\boldsymbol{r}, \boldsymbol{r}^{(1)}) = \begin{cases} -K/(L \cdot g), & \text{if } \nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r}) > 0 \\ K/(L \cdot g), & \text{if } \nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r}) < 0 \end{cases} . \tag{6}$$

In Fig. 4(b), we draw the distribution of the opposite of the new weight value (6). It is almost constant, for cubes with same size and dielectric configuration. Thus, the new FRW scheme may exhibit much less variance.

To remove the variance of new weigh value in Fig. 4(b) and brought by L in (6) further, the stratified sampling (SS) technique is employed. We divide the cube's surface into two stratas, one with positive weight value and the other with negative weight value. And, the Gaussian surface is decomposed so that each face is a strata. Take a cubic Gaussian surface as an example. With the SS technique, the original integral (2) is now calculated separately as 12 integrals. The combination of IS and SS techniques delivers a comphrensive scheme of variance reduction to the FRW algorithm. Numerical experiments demonstrate it brings **3X** or more speedup without memory overhead and the loss of accuracy [5]. The scheme is also superior to the techniques in [4].



(a) $\dfrac{\nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})}{P(\boldsymbol{r}, \boldsymbol{r}^{(1)})}$ (b) $\dfrac{K \cdot \nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})}{\left|\nabla_r P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \bullet \hat{n}(\boldsymbol{r})\right|}$

Fig. 4 The values in WVT for $\boldsymbol{r}^{(1)}$ on top and side faces of the cube. (a) for the ω of the standard FRW, (b) for the $\tilde{\omega}$ using the importance sampling.

3.3 Fast space management for chip-scale structures

For each hop of a walk, a conductor-free cubic transition domain is constructed. We need to find the distance from current point to its nearest conductor block to make the transition cube as large as possible. When there are a large number of conductor blocks, an Octree based space management structure [5], which indexes all the blocks, should be used. For each node in Octree, the "candidate list" of the nearest blocks for points in the node is generated. So, only distances to the fewer blocks in candidate list are calculated for each FRW hop, which is indispensable for preserving the FRW algorithm's efficiency while extracting chip-scale large structures.

Construction of the Octree can be done by inserting all conductor blocks into an empty Octree's root one by one. Two thresholds: n_t and l_t, are used. When a node's candidate list contains more than n_t blocks, the node should be divided into 8 subnodes. To avoid the tree growing too high, the spatial dimension of a leaf node should not less than l_t. The primary operation for constructing the Octree is checking if a block should be added into a node's candidate list. This is judged with the domination relationship defined as follows:

Definition 1: T is a node, and B_1, B_2 are two blocks. If for any point $P \in T$, and $P \notin B_1 \cup B_2$, $d(P, B_1) \le d(P, B_2)$, we say B_1 *dominates* B_2 regarding T.

A block will not be inserted into a node's candidate list unless it's not dominated by any other blocks in that candidate list. Thus, the domination relationship should be judged between the block and every block in the node's candidate list. This could cost huge time while handling a structure with thousands of blocks. Techniques are proposed to reduce the domination judgment.

Definition 2: The *size* of a cuboid node is defined as the maximum of the node's length, width and height.

Definition 3: The *distance limit* $L(T)$ of an Octree node T is the minimum distance between it and a block in its candidate list, plus the node's size.

Actually, $L(T)$ is an upper bound of the distance to nearest block from any point in node T. For a new block B, the candidate list of a leaf node T and block B_0 in the list, if $d(B, T) \ge L(T) = d(B_0, T) + Size(T)$, B is dominated by B_0. Because for any point P in the conductor-free space within T, we have $d(P, B_0) \le d(T, B_0) + Size(T) = L(T) \le d(T, B) \le d(P, B)$. While inserting a block to the

candidate list, the value of distance limit dynamically changes. Algorithm 1 describes the candidate checking procedure. For most checked blocks, it returns at the beginning instead of after testing all blocks in the list.

Algorithm 1 CandidateCheck(block B, node T)

```
1.  d := d(B, T); l is the size of T;
2.  If d ≥ L(T) then return false;
3.  For each b in the candidate list of T do
4.      If b dominate B then return false;
5.      Elseif B dominate b then
6.          Remove b from the candidate list of T;
7.      Endif
8.  Endfor
9.  Add B to the candidate list of T;
10. If (d + l) < L(T) then L(T) := d + l; Endif
11. Return true.
```

With the distance limit, the time for constructing the Octree can be reduced from half an hour to about 3 seconds (**600X**) for a case with 37062 conductor blocks. The construction of Octree can be further accelerated with the idea of searching only a *neighbor region*, and by using an Octree-grid hybrid structure, rather than the pure Octree structure [12]. The techniques also reduce the time for performing FRW procedure by **2X**.

4. Usage of RWCap and Examples

The program set RWCap includes two programs: TechGFT written in Matlab, and RWCap-solver written in C++. Now, we only share their executable versions.

TechGFT generates the binary files storing GFTs and WVTs for transition cubes with specified two-layer dielectric configurations. The input is a text file with ".diel_conf" suffix. Each line in it follows the syntax:

.d <down_diel> <up_diel>

It indicates a dielectric layer interface, where "down_diel" and "up_diel" are two numbers meaning the relative dielectric permittivities below and above the interface, respectively. Once the input file with dielectric configurations is ready, executing "RunTechGTF" under the current path in Matlab starts the generation of GFT and WVT files. It will prompt specifying a ".diel_conf" file, and output the data files in a folder with the same name as the ".diel_conf" file.

RWCap2 has two running modes. The first one:

$ rwcap2 --gft <data_dir>

registers folder "data_dir" storing the data generated by TechGFT. It is omitted if solving a single-dielectric case. Then, the second mode performs FRW based extraction:

$ rwcap2 -f <file_name> [-p <rel_self_cap> | -t <num_walks>] [-n <num_thread>] [-w min_width>]

The "-f" argument indicates the ".qbem" format file describing the 3-D structure for capacitance extraction. The next two arguments specify the convergence criterions. The last two specify the number of threads used for multi-core parallel computing and the minimum wire width for optimal space management, respectively. Despite the last four arguments are optional, the second one is usually specified to ensure the accuracy.

In the package of RWCap (Version 1), the ".qbem" files for cases in [5] are in folder "cases". In RWCap2 package, folder "LargeCase" includes 4 large structures [12]. Fig. 5 shows the second case (called "FreeCPU") with 3037 nets formed by 37062 conductor blocks. And, the largest one has half a million conductor blocks. Each of these ".qbem" files describes layered medium configuration and the geometries of conductor blocks.

On a Linux server with Intel Xeon E5-2650 8-core 2.0 GHz CPU, the parallelized RWCap2 is able to extract capacitances for all nets in "FreeCPU" in 35.2 minutes. This means an average extraction speed of about **0.7 second/net**, under the accurate 1% error criterion.

Acknowledgments

Several students in my group contributed to the development of RWCap program set. Among them, Hao Zhuang was the major developer of RWCap (Version 1) and TechGFT, and Chao Zhang was the major developer of RWCap2. Other contributors include Gang Hu, Zhi Liu and Ting Dai. Their efforts are greatly appreciated.

References

[1] Y. Le Coz and R. B. Iverson, *Solid State Electron.*, 35, p. 1005 (1992).

[2] A. Brambilla and P. Maffezzoni, *IEEE Microwave Guided Wave Letters*, 10, p. 304 (2000).

[3] M. Kamon, R. B. Iverson, in *EDA for IC Implementation, Circuit Design, and Process Technology*, L. Lavagno, L. Scheffer, G. Martin (ed.), CRC Press, 2006

[4] S. H. Batterywala, M. P. Desai, *International Conference on VLSI Design*, p. 85 (2005).

[5] W. Yu, H. Zhuang, C. Zhang, G. Hu, Z. Liu, *IEEE Trans. Computer-Aided Design*, 32, p. 353 (2013).

[6] K. Zhai, W. Yu, H. Zhuang, *Design, Automation & Test in Europe Conference*, p. 1661 (2013).

[7] K. Nabors, J. White, *IEEE Trans. Computer-Aided Design*, 10, p. 1447 (1991).

[8] W. Yu, X. Wang, Z. Ye, Z. Wang, *IEEE Trans. Computer-Aided Design*, 27, p. 1508 (2008).

[9] T. A. El-Moselhy, I. M. Elfadel, L. Daniel, *International Conference on Computer-Aided Design*, p. 662 (2008).

[10] ————, *International Conference on Computer- Aided Design*, p. 752 (2009).

[11] RWCap: a 3-D floating random walk based capacitance solver. http://learn.tsinghua.edu.cn:8080/2003990088/rwcap.htm

[12] C. Zhang and W. Yu, *IEEE Trans. Computer-Aided Design*, 32, p. 1633 (2013).

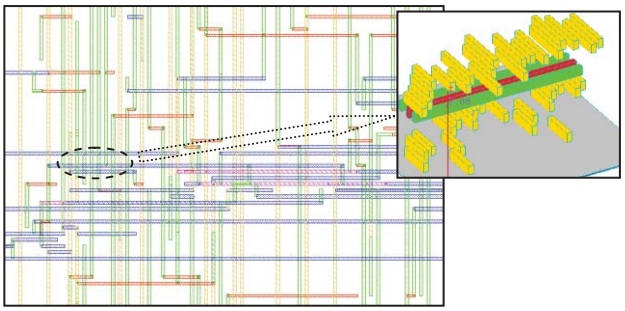


Fig. 5 Partial 2-D layout of "FreeCPU" case, and its zoom-in.