

Applications of Monte Carlo Method to 3-D Capacitance Calculation and Large Matrix Decomposition

[Invited Paper]

Wenjian Yu

TNList, Dept. Computer Science and Tech., Tsinghua University, Beijing 100084, China.

Email: yu-wj@tsinghua.edu.cn

Abstract

With the increasing demands of performance, yield, and turn-around time of smart device design, fast and accurate three-dimensional (3-D) numerical simulation has become indispensable and poses the computational challenge to the designers. On the other hand, the Monte Carlo (MC) method has the advantages of better parallelism, better scalability for very large structures, tunable accuracy, etc., and therefore gains more and more attentions as an effective computing scheme. In this paper, the MC based techniques for 3-D capacitance calculation are presented, with focus on the applications in integrated circuit design and touchscreen design. The MC based approach for partial singular value decomposition (SVD) of large matrices is also presented, which could benefit various applications of big-data analytics.

1. Introduction

Monte Carlo (MC) method is one of “the top 10 algorithms in the 20th century” [1]. Instead of the random process used to investigate a system’s behavior, the MC method in this paper is referred to as a computing method for deterministic or stochastic quantities. For solving partial differential equation, a kind of MC method called random walk method or Green’s function MC method has been developed. It has the following advantages if compared with deterministic methods:

- Locality. It allows to obtain the solution at few local positions without solving the whole equations.
- Stability of accuracy. Its accuracy is more stable and tunable, as its error is mainly stochastic error.
- Geometric adaptability. Without geometry discretization, it reduces memory and avoids preprocessing.
- Suitability for large problem. Without generation of linear equation system, it works for large problem.
- Natural parallelism, due to independent samplings.

The drawbacks of the MC method are mainly the generality and the computational speed. It relies on the stochastic explanation of the original problem’s solution. So, it is only able to solve limited kind of equations, not as general as the finite difference or finite element methods. Its runtime is inversely proportional to the square of error. So, for the problem where we need the whole solutions or higher accuracy the MC method would run much slower than the deterministic methods. It is commonly agreed that it is most efficient when point

This work is supported by National Natural Science Foundation of China (No. 61422402).

values or linear functionals of the solution are needed [2].

With the increasing demands for 3-D, large-scale simulation during the design of integrated circuit (IC), flat panel display (FPD) and other kind of smart devices, the deterministic methods are faced by large challenge of computation cost and even accuracy. The MC based approaches regain the attraction due to the population of parallel computing infrastructures. Actually, they have become the major solution for some practical problems, like the interconnect capacitance field solver for the design of vary large scale integrated (VLSI) circuits.

In this paper, the theory and recent developments of the MC based techniques for large-scale simulation and computation problems are surveyed. We first present the random walk method for calculating electric potential and capacitances, and its connection to a Markov random process. Then, the extensions for tackling the challenges in simulating the touchscreen structures (a special kind of FPD) are presented. Lastly, we move forward to a MC based matrix decomposition scheme which is applicable to large-scale machine learning applications.

2. The Basics of Monte Carlo method

The modern MC method is a kind of method which utilizes a process of random sampling with the aid of computer generated randomness. It was invented for estimating quantities in controlled fission and thermonuclear reactions in 1940s [3]. Consider a simple problem of calculating an integral: $I = \int_0^1 f(x)dx$. We derive

$$I = \int_0^1 P(x) \frac{f(x)}{P(x)} dx, \quad (1)$$

where $P(x)$ is a probability density function on interval $[0, 1]$. So, I can be interpreted as the statistical mean of the random quantity $f(x)/P(x)$ if random variable x follows the distribution $P(x)$. $P(x)$ can be arbitrary, but a trivial choice is the uniform distribution ($P(x) \equiv 1$). So, approximating the stochastic mean with the average of N sample values,

$$I = \int_0^1 f(x)dx \approx \tilde{I} = \frac{1}{N} \sum_{i=1}^N f(x_i), \quad (2)$$

where x_i is the i -th uniform random sample on $[0, 1]$. For a sufficient large N , \tilde{I} in (2) can attain enough accuracy. This is the MC process for calculating an integral. It is advantageous over the conventional numerical quadratures for high-dimensional integrals.

The central limit theorem tells us that \tilde{I} is a Gaussian random quantity as N approaches to the infinity, since it is the average of independent random variables with

identical distribution. The standard deviation (Std) of this Gaussian distribution measures the error of \bar{I} , which is often called 1- σ error. It can be estimated by the variance of sampled values $\{f(x_i)\}$, and is thus tunable during the MC process. We can monitor the error and terminate the computation once it attains a preset accuracy within a certain level of confidence.

Similarly, the MC process applies to the linear algebra computations. Take a summation problem as an example.

$$S = \sum_{i=1}^m a_i = \sum_{i=1}^m [p_i \left(\frac{a_i}{p_i}\right)], \quad (3)$$

where $\{p_i\}_{i=1}^m$ is a set of probability. Choosing a simple uniform probability, i.e. $p_i \equiv 1/m$, we have

$$S = \sum_{i=1}^m \left[\frac{1}{m} (m \cdot a_i)\right] \approx \bar{S} = \frac{1}{N} \cdot m \sum_{j=1}^N a_{i_j}, \quad (4)$$

where i_j is the j -th uniform sample among indices from 1 to m . This method for calculating S is the basis of the MC method for other linear algebra problems [4, 5].

Two major concerns of the MC method is how to make each random sample, and how to reduce the number of samples N for a given accuracy. Notice in practice that the sampling probability function $P(x)$ in (1) and $\{p_i\}_{i=1}^m$ in (3) can be arbitrary. The techniques of rejection sampling and Markov chain Monte Carlo (MCMC) can be used for generating samples with complicated distribution [6, 7]. As for reducing N , the variance reduction techniques like importance sampling (IS) and stratified sampling are often used, which construct special $P(x)$ or $\{p_i\}_{i=1}^m$ to accelerate the convergence of the MC process [7].

3. The Monte Carlo based capacitance calculation

In this section, we first present the theory of the floating random walk algorithm for capacitance extraction. Then, recent developments and related topics are addressed.

3.1 The floating random walk algorithm for the capacitance extraction in VLSI design

With the feature size scaled down to nanometers, the signal delay on interconnect wires has dominated the total circuit delay of IC. Therefore, modeling the interconnects becomes a critical task in the design of digital VLSI circuits. The capacitance extraction refers to calculating the capacitance parameters in the equivalent RC circuit of interconnects, which is the base of accurate circuit modeling and simulation. Capacitance relates the electric potential and the induced charge of conductors. It depends on the 3-D electrostatic field around the considered wire segment. So, electrostatic field solver is desirable for capacitance extraction, and provides calibration for other empirical approaches [7]. The field-solver method can be based on deterministic finite difference, finite element and boundary element methods, or the floating random walk (FRW) method.

In 1970s or earlier, the random walk (or MC) method had been used for calculating electric potential [8], but was not for capacitance extraction until 1992 [9]. The random walk method is based on expressing the electric

potential $\phi(\mathbf{r})$ at point \mathbf{r} as an integral of the potential on surface $S^{(1)}$ enclosing \mathbf{r} :

$$\phi(\mathbf{r}) = \oint_{S^{(1)}} P_{RW}(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) dS^{(1)}, \quad (5)$$

where $P_{RW}(\mathbf{r}, \mathbf{r}^{(1)})$ is called surface Green's function and can be regarded as a probability density function.

Without loss of generality, we first consider a problem of calculating $\phi(\mathbf{r})$ within the domain shown in Fig. 1. On the domain boundary, the potential is zero except on Γ_1 . Eq. (5) suggests that $\phi(\mathbf{r})$ is the stochastic mean of $\phi(\mathbf{r}^{(1)})$, and can be calculated by randomly sampling $S^{(1)}$ with probability function P_{RW} . If at the sample point $\phi(\mathbf{r}^{(1)})$ is unknown, we construct a similar surface $S^{(2)}$ enclosing $\mathbf{r}^{(1)}$ and sample it. Repeat this until we reach a point with known potential, which is an estimate for $\phi(\mathbf{r})$. This spatial sampling procedure forms a random walk (including some hops). With N random walks performed, the average of returned electric potentials approximates $\phi(\mathbf{r})$. This is the random walk method.

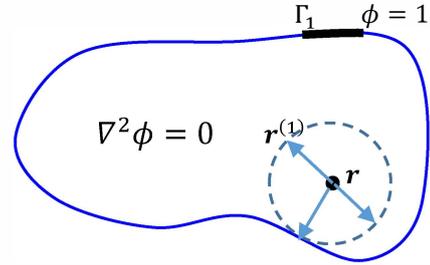


Figure 1. The illustration of random walk method for calculating electric potential.

Based on the problem's geometry, we can define a dual problem. Suppose particles are released at point \mathbf{r} . With same manners of generating the closed surface $S^{(i)}$, the particle randomly hops from its current position to $S^{(i)}$. The end point $\mathbf{r}^{(i)}$ of a hop only depends on its start point $\mathbf{r}^{(i-1)}$ and is determined by a Markov transition probability density function $P_{MT}(\mathbf{r}^{(i-1)} \rightarrow \mathbf{r}^{(i)})$. If we set $P_{MT}(\mathbf{r}^{(i-1)} \rightarrow \mathbf{r}^{(i)})$ identical to $P_{RW}(\mathbf{r}^{(i-1)}, \mathbf{r}^{(i)})$ in the random walk process, the particle's trajectory is the same as the spatial sampling trajectory in the latter, which is a well-defined Markov process. In the dual problem, the probability of a particle released at \mathbf{r} finally reaching Γ_1 : $Pr(\mathbf{r})$ is concerned. Due to the property of Markov process,

$$\begin{aligned} Pr(\mathbf{r}^{(i-1)}) &= \oint_{S^{(i)}} P_{MT}(\mathbf{r}^{(i-1)} \rightarrow \mathbf{r}^{(i)}) Pr(\mathbf{r}^{(i)}) dS^{(1)} \\ &= \oint_{S^{(i)}} P_{RW}(\mathbf{r}^{(i-1)}, \mathbf{r}^{(i)}) Pr(\mathbf{r}^{(i)}) dS^{(1)}. \end{aligned} \quad (6)$$

This equation is just the same as that which defines the relationship among electric potentials (5), if we replace $Pr(\mathbf{r}^{(i)})$ with $\phi(\mathbf{r}^{(i)})$. So, calculating $\phi(\mathbf{r})$ equals to calculating $Pr(\mathbf{r})$, which by definition can be obtained by a MC process. That means we emulate the random walks of particles and count the probability of particles finally reaching Γ_1 . This equivalence between $\phi(\mathbf{r})$ and $Pr(\mathbf{r})$ explains the convergence of the random walk method. For the discussion about problem with more general boundary conditions, please refer to [10].

For calculating the capacitances related to a specified conductor j (often called *master conductor*), a Gaussian surface G_j should be constructed to enclose it (see Fig. 2). With the Gauss theorem, conductor j 's electric charge $Q_j = \oint_{G_j} F(\mathbf{r})g \oint_{S^{(1)}} \omega(\mathbf{r}, \mathbf{r}^{(1)})q(\mathbf{r}, \mathbf{r}^{(1)})\phi(\mathbf{r}^{(1)})ds^{(1)} ds$, (7) where $F(\mathbf{r})$ is the dielectric permittivity at point \mathbf{r} , $q(\mathbf{r}, \mathbf{r}^{(1)})$ is the probability density function for sampling $S^{(1)}$, the surface of a transition domain. g is a constant, which satisfies $\oint_{G_j} F(\mathbf{r})gds = 1$. $q(\mathbf{r}, \mathbf{r}^{(1)})$ may be different from the surface Green's function, and $\omega(\mathbf{r}, \mathbf{r}^{(1)})$ is the weight value [7]. Thus, Q_j can be estimated as the statistical mean of sampled values on G_j , which is further the mean of sampled electric potentials on $S^{(1)}$ multiplying the weight value. The spatial sampling procedure for calculating the potential forms a floating random walk (FRW) including a sequence of hops. Usually, each hop is from the center of a transition domain to its boundary. With a number of such walks, the statistical mean of the weight values for the walks terminating at conductor i approximates the capacitance C_{ji} between conductors i and j (if $j \neq i$), or the self-capacitance C_{jj} of master conductor j .

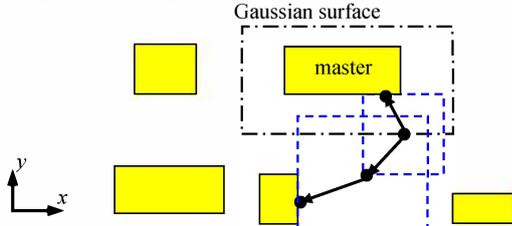


Figure 2. Two examples of random walk in the FRW method for capacitance extraction (a 2-D top view).

Although the surface Green's function for a spherical transition domain (as that in Fig. 1) has simple analytical expression [11], the cubic transition domain is widely adopted because it well fits the IC layout including mostly Manhattan shapes. This means larger probability for terminating a walk quickly. The sampling probability and weigh value for a cubic domain can be numerically pre-calculated and tabulated, so as to largely accelerate the sampling operation. This induces some discretization, but its error is tolerable for capacitance extraction.

The runtime of the FRW method is proportional to the product of the number of random walks, the average number of hops per walk and the average time for performing a hop. The variance reduction techniques can reduce the number of walks [7, 12]. The placement of the Gaussian surface also affects. For a 3-D IC structure including cylindrical through-silicon vias, the treatment of the Gaussian surface and the corresponding IS can bring 10X speedup [13]. For large structure including many conductors, employing an efficient space management technique is crucial for reducing the time for performing a hop. For structures including only *Manhattan* conductors, a major idea is to maintain a candidate list of conductor cuboids for each spatial cell,

such that for any point in the cell its nearest conductor is in the list. Therefore, the inquiry of nearest conductor only demands to traverse the candidate list and can be executed very quickly. This was recently extended to handle general non-Manhattan conductors [14].

3.2 Recent developments and related topics

The touch panel has been combined with FPD to largely enhance the interactivity and user experience of various customer electronics. Most of these touchscreens utilize the capacitive touch sensor (see Fig. 3), because of its advantages in durability, reliability and capability. To validate the functionality (like Multi-Touch, Force-Touch) and sensitivity of the touchscreen, calculating the relevant capacitances becomes an important and frequent task during the design of high-quality touchscreens. This is similar to the capacitance extraction in the design of VLSI circuits, but with distinct differences (see Table 1).

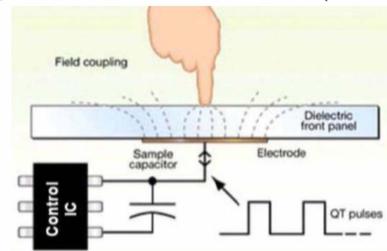


Figure 3. The illustration of capacitive touch panel.

Table 1. The differences between capacitance extraction for VLSI circuit and capacitance calculation for touchscreen design.

	VLSI circuit capacitance extraction	Touchscreen capacitance calculation
Conductor geometry	Mostly Manhattan shape, with moderate aspect ratio	Generally non-Manhattan shape, with very large aspect ratio
Dielectric environment	On-chip dielectric insulators; relatively fixed dielectric profile	In-device dielectrics and out-device air; arbitrary dielectric configuration
Accuracy demand	Mainly self-capacitance for delay calculation	Need accurate coupling capacitances

Because there are divergent process technology recipes for the touchscreen, it is desirable to have a unified dielectric pre-characterization scheme instead of pre-calculating the FRW transition tables for each process technology [7]. Also note that inclusion of air leads to a large range of dielectric permittivities. This prevents the existing unified pre-characterization technique [15] from being applied. In [16], we proposed a unified dielectric pre-characterization scheme which overcomes these difficulties. We also implemented the algorithm on a large-scale computer cluster. With algorithm reconstruction and MPI parallel computing, we achieved 93X~114X speedup using 120 processors. This makes an efficient capacitance simulator for touchscreen design.

To leverage the tremendous computational power of the general-purpose graphics processing units (GPUs), the GPU-friendly algorithmic flow and data structure have been investigated for the FRW based capacitance

extraction [17, 18]. The major efforts have been paid to minimize the divergence of operations and alleviate the bottleneck of global memory of GPUs. Another recent advancement is the combination of the FRW scheme and a Markov-chain random walk scheme to accelerate the capacitance extraction, and handle circuits including intellectual-property (IP) protected or geometry-complex substructures [19]. The resulted macromodel-aware algorithm enhances the capability of the state-of-the-art FRW based capacitance solver with negligible overhead.

Statistical modeling and simulation is required to capture the uncertainties in the nanometer manufacturing process. For capacitance extraction, the geometric variation of interconnect wire is of major concern. Although some techniques were proposed for statistical capacitance extraction [20, 21], the MC method is still the golden standard, and is the only choice for the situation where a lot of independent variables are involved. Except for capacitance, the calculation of resistance and inductance/impedance [22] are also required. The FRW method can be applied to resistance calculation, but with less advantage over deterministic methods. In contrast, MC based impedance extraction is still an open problem. Although there is a recent progress on solving the telegraph equation [23], applying the MC method to a general wave equation is difficult.

4. The Monte Carlo based large matrix approximation

Low-rank matrix factorization plays a crucial role in data analysis and scientific computing. It accelerates the solution of linear equation system, or make a choice of model order reduction for the simulation problems in electronic design [7, 24]. The optimal rank- k approximation of matrix A can be obtained by its SVD:

$$A \approx A_k = U_k \Sigma_k V_k^T, \quad (8)$$

where U_k and V_k include the k dominant left and right singular vectors respectively, and the diagonal matrix Σ_k has the k largest singular values. This approximation has the minimal error $\|A_k - A\|$ among all rank- k matrices.

Recent years have witnessed a boom in “Big Data” related research. Large-scale data analytics requires machine learning approaches, which often involve matrix SVD, or principal component analysis (PCA). Efficiently approximating large matrices is of large importance. To tackle the challenges caused by the large computational cost using traditional matrix methods, the MC based techniques have been proposed for these linear algebra problems [4, 25-28]. The results show that randomization can be a powerful computational resource to develop algorithms with improved runtime and stability properties.

An example scheme of randomized matrix decomposition is shown in Fig. 4 [26, 27]. It uses random sampling to identify the subspace capturing the dominant actions of matrix A (represented by orthonormal matrix Q). Then, standard factorizations are performed on the smaller B , which finally make the low-rank factorizations of A . The

scheme is adaptive to parallel computing environments and orders of magnitude faster than traditional methods.

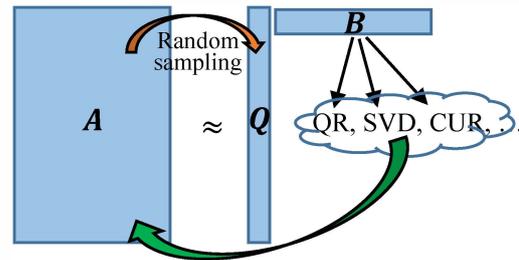


Figure 4. A scheme of randomized matrix decomposition.

5. Conclusion

The Monte Carlo method has diverse usages nowadays, due to its unique advantages over deterministic methods. It is expected to have more successful applications to large-scale simulation and machine learning problems.

References

- [1] J. Dongarra and F. Sullivan, *Computing in Science & Engineering*, p. 22 (Jan. 2000).
- [2] C.-O. Hwang, J. A. Given and M. Mascagni, *Journal of Computational Physics*, 174, p. 925 (2001)
- [3] N. Metropolis and S. Ulam, *Journal of the American Statistical Association*, 44, p. 335 (1949).
- [4] S. Eriksson-Bique, M. Solbrig, M. Stefanelli, et al., *SIAM J. Sci. Comput.*, 33, p. 1689 (2011).
- [5] H. Ji, M. Mascagni, and Y. Li, *SIAM J. Numer. Anal.*, 51, p. 2107 (2013).
- [6] C. Zhang and W. Yu, *IEEE ASP-DAC*, p. 756 (2014).
- [7] W. Yu and X. Wang, *Advanced Field-Solver Techniques for RC Extraction of Integrated Circuits*, Springer Inc., 2014.
- [8] G. M. Royer, *IEEE Trans. MTT*, 19, p. 813 (1971).
- [9] Y. Le Coz and R. B. Iverson, *Solid State Electron.*, 35, p. 1005 (1992).
- [10] R. M. Bevensee, *Proc. IEEE*, 61, p. 423 (1973).
- [11] A. Brambilla and P. Maffezzoni, *IEEE Microw. Guided Wave Lett.*, 10, p. 304 (2000)
- [12] S. H. Batterywala and M. P. Desai, *IEEE VLSID*, p. 85 (2005).
- [13] C. Zhang, W. Yu, Q. Wang, et al., *IEEE Trans. Comput.-Aided Design*, 34, p. 1977 (2015).
- [14] Z. Xu, C. Zhang and W. Yu, *IEEE Trans. Comput.-Aided Design*, *accepted* (2016).
- [15] G. Rollins, *Rapid3D 20X performance improvement*, 2010.
- [16] Z. Xu, W. Yu, et al., *ACM GLSVLSI*, p. 99 (2016).
- [17] K. Zhai, W. Yu and H. Zhuang, *IEEE DATE*, p. 1661 (2013).
- [18] N. D. Arora, S. Worley, and D. R. Ganpule, *IEEE EDSSC*, p. 459 (2015).
- [19] W. Yu, B. Zhang, et al., *IEEE DATE*, p. 1225 (2016).
- [20] W. Yu, Q. Zhang, Z. Ye, et al., *Microelectronics Reliability*, 52, p. 704 (2012).
- [21] T. A. El-Moselhy, I. M. Elfadel and L. Daniel, *IEEE ICCAD*, p. 662 (2008).
- [22] X. Wang, D. Liu, W. Yu, et al., *IEICE Trans. Electronics*, E88-C, p. 232 (2005).
- [23] J. A. Acebron and M. A. Ribeiro, *Journal of Computational Physics*, 305, p. 29 (2016).
- [24] J. R. Phillips and L. M. Silveira, *IEEE Trans. Comput.-Aided Design*, 24, p. 43 (2005).
- [25] P. Drineas, R. Kannan and M. W. Mahoney, *SIAM J. Comput.*, 36, p. 132 (2006).
- [26] N. Halko, P.-G. Martinsson and J. A. Tropp, *SIAM Review*, 53, p. 217 (2011).
- [27] Y. Gu, W. Yu and Y. Li, *arXiv:1606.09402v2* (2016).
- [28] P. Drineas and M. W. Mahoney, *Communications of the ACM*, 59(6), p. 80 (2016).