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

Wenjian Yu
Department of Computer Science & Tech.
Tsinghua University
Beijing, China

Oct. 26, 2016

Outline

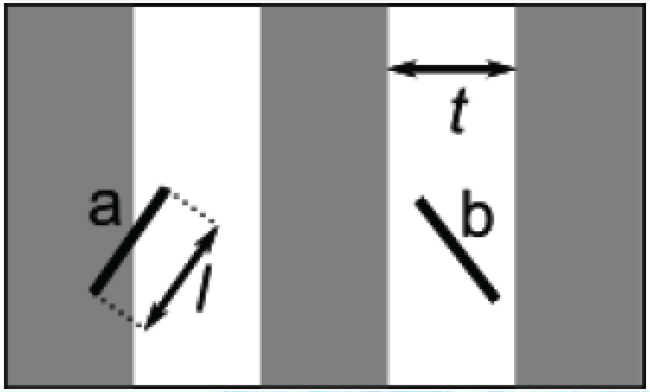
- Introduction
- Basics of Monte Carlo method
- MC based capacitance calculation
- MC based large matrix approximation
- Conclusion

Introduction

- Definition
  - **Monte Carlo methods** (or Monte Carlo experiments) are a broad class of computational algorithms that rely on repeated *random sampling* to obtain numerical results

WIKIPEDIA

  - We refer to it as a computing method for deterministic or stochastic quantities, instead of the random process for imitating a complex system's behavior
  - A historical example: Buffon's Needle Problem (1777)
    - Drop a needle on a lined surface
    - $\pi \approx 2n/m$, where n is the count of experiments, m is count of intersection of needle and grid

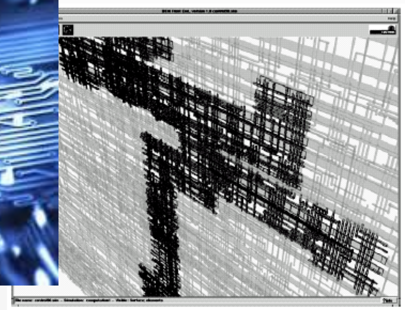


Introduction

- MC method for solving partial differential equation
  - Called random walk method, or Green's function MC
  - Advantages as compared with deterministic methods
    - Locality: Obtain the solution at a local position
    - Accuracy stability: Mainly stochastic error
    - Geometric adaptability: No geometry discretization
    - Scalability for large problem: Low memory w/o building equ.
    - Natural parallelism: Independent samplings
  - Drawbacks
    - The generality: rely on a stochastic explanation
    - Computational speed: slow convergence rate ($T \propto \frac{1}{\mathrm{error}^2}$)
  - It's most efficient when point values or linear functionals of the solution are needed

Introduction

- Challenges of 3-D and large-scale simulation



  - Large computational time, and even error
- MC method regains the attraction
  - Due to the popularity of parallel computing infrastructures
  - Beats deterministic methods in some applications
    (orders of magnitude faster than fast BEM for capacitance extraction of large IC structures)
  - Also find applications in large-scale linear algebra computations, useful for Big-Data analytics

[1] **W. Yu**, et al., "... … in designing flat panel displays," ***Journal of the Society for Information Display***, 2016

[2] **P. Drineas** and **M. W. Mahoney**, "RandNLA: randomized numerical linear algebra," ***Communications of the ACM***, 2016

Introduction

- In this talk
  - Survey on theory and recent development of the MC based techniques for large-scale simulation and computation
  - The probabilistic potential theory for the random walk method for electrostatic PDE
  - MC based techniques for 3-D capacitance calculation
    - The floating random walk method for the capacitance extraction in VLSI design
    - Recent enhancement for tackling the challenges in simulating the touchscreen structures, and related topics
  - MC based technique for large matrix approximation

Outline

- Introduction
- Basics of Monte Carlo method
- MC based capacitance calculation
- MC based large matrix approximation
- Conclusion

Basics of Monte Carlo method

- Example 1 -- Integration

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

  - $P(x)$ is a probability density function on [0, 1]
  - A stochastic explanation:
    - Random variable $\xi \sim P(\xi)$
    - Sample value: $f(x_i)/P(x_i)$

$$I = \mathbb{E}\left(\frac{f(\xi)}{P(\xi)}\right) \approx \tilde{I} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{P(x_i)}$$

$\tilde{I}$ attains enough accuracy with large N

  - With the central limit theorem
    - $\tilde{I} \sim \mathrm{N}(I, \sigma^2)$. So, σ measures error of $\tilde{I}$ (often called 1-σ error)

$$\sigma \equiv \sqrt{\mathrm{var}(\tilde{I})} = \sqrt{\mathrm{var}(f(\xi)/P(\xi))/N},$$

$$\mathrm{var}\left(\frac{f(\xi)}{P(\xi)}\right) \approx \frac{1}{N-1}\sum_{i=1}^{N}\left[\frac{f(x_i)}{P(x_i)} - \tilde{I}\right]^2$$

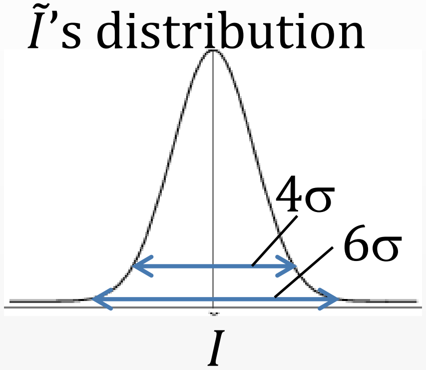
Error $\propto 1/\sqrt{N} \approx \frac{1}{\sqrt{\mathrm{Time}}}$

Error can be estimated during the MC process

Basics of Monte Carlo method

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

Error $\propto 1/\sqrt{N}$



It is superior to conventional numerical quadratures for high-dimensional integral

- Example 2 -- Linear algebra
  - $S = \sum_{i=1}^{m} a_i$ Define probabilities $\{p_i\}$ for index i, $\sum p_i = 1$
  - $S = \sum p_i (a_i / p_i)$, i.e., S is the statistical mean of $\frac{a_i}{p_i}$, if index i is chosen with probability p_i. ⟶ $\approx \tilde{S} = \frac{1}{N} \cdot \sum_{j=1}^{N} \frac{a_{i_j}}{p_{i_j}}$
  - Similar method applies to $S = \sum_{i=1}^{m} a_i x_i$
  - It's the basis of the MC method for linear algebra problems (linear equation system [1], random walk based circuit simulation [2], et al.)

[1] **H. Ji**, **M. Mascagni**, and **Y. Li**, "Convergence analysis of Markov Chain …," ***SIAM J. Numer. Anal.***, 2013
[2] **H. Qian, S. Nassif,** and **S. Sapatnekar**, "Power grid analysis using random walks," ***IEEE Trans. CAD***, 2005

Basics of Monte Carlo method

- The key point of modern MC: *using the random sampling process with the aid of computer generated randomness*
- Concerns for developing an efficient MC method
  - Efficient pseudo-random number generator
  - How to make random sample following arbitrary distribution?
    - Rejection sampling
    - Markov chain Monte Carlo
  - How to reduce the number of samples for a preset accuracy?
    - Variance reduction (importance sampling, stratified sampling, …)
    - Construct special $P(x)$ or $\{p_i\}_{i=1}^{m}$ to accelerate convergence

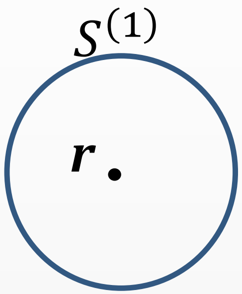
[1] **I. Beichl** and **F. Sullivan**, "The metropolis algorithm," ***Computing in Science & Engineering***, 2000
[2] **W. Yu** and **X. Wang**, ***Advanced Field-Solver Techniques for RC Extraction of Integrated Circuits***, 2014.
[3] **C. Zhang** and **W. Yu**, "Efficient techniques … using floating random walk algorithm," ***Proc. ASP-DAC***, 2014

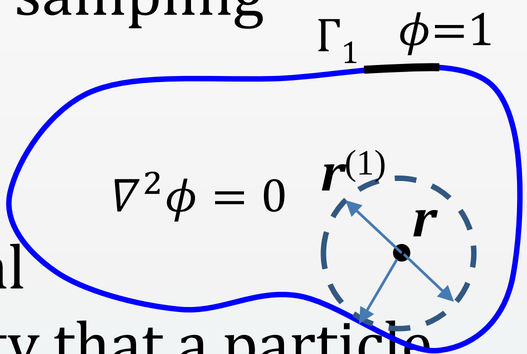
Outline

- Introduction
- Basics of Monte Carlo method
- MC based capacitance calculation
- MC based large matrix approximation
- Conclusion

MC Based Capacitance Calculation

- The random walk method
  - Electric potential $\phi(\boldsymbol{r}) = \oint_{S^{(1)}} P_{RW}(\boldsymbol{r}, \boldsymbol{r}^{(1)})\phi(\boldsymbol{r}^{(1)})\mathrm{d}s^{(1)}$
  - $P_{RW}(\cdot)$ is a PDF, i.e. surface Green's function
  - It derives a RW method for calculating $\phi(\boldsymbol{r})$: sampling on $S^{(1)}, S^{(2)} \ldots, S^{(k)}$ until $\phi\left(\boldsymbol{r}^{(k)}\right)$ is known
  - Explain its convergence with a dual problem:
  - Particles released at $\boldsymbol{r}$, following same spatial transitions as the RW. What is the probability that a particle reaches Γ_1?





A Markov process defined by $P_{MT}\left(\boldsymbol{r}^{(i-1)} \to \boldsymbol{r}^{(i)}\right)$

$$P_{MT}(\boldsymbol{r}^{(i-1)} \to \boldsymbol{r}^{(i)}) = P_{RW}(\boldsymbol{r}^{(i-1)}, \boldsymbol{r}^{(i)}) \qquad \Pr(\boldsymbol{r}^{(i-1)}) = \oint_{S^{(i)}} P_{MT}(\boldsymbol{r}^{(i-1)} \to \boldsymbol{r}^{(i)})\Pr(\boldsymbol{r}^{(i)})\mathrm{d}s^{(1)}$$

⇒ $\phi(\boldsymbol{r}) = \Pr(\boldsymbol{r})$, which by definition is got with a converged MC process

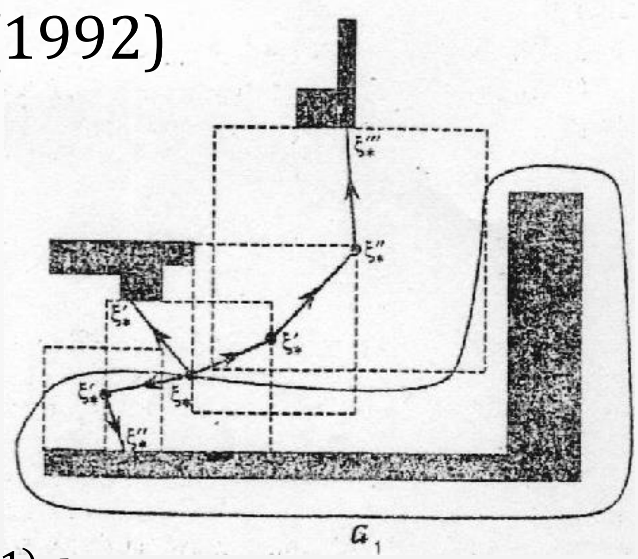
  - It applies to general boundary settings

[1] **R. Bevensee**, "Probabilistic potential theory applied to electrical engineering problems," ***Proc. IEEE***, 1973

MC Based Capacitance Calculation

- The floating random walk method (1992)
  - Definition of Capacitance $\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{12} & C_{22} & C_{23} \\ C_{13} & C_{23} & C_{33} \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$

$$Q_1 = C_{11}V_1 + C_{12}V_2 + C_{13}V_3$$



  - Gauss theorem for electrostatic field

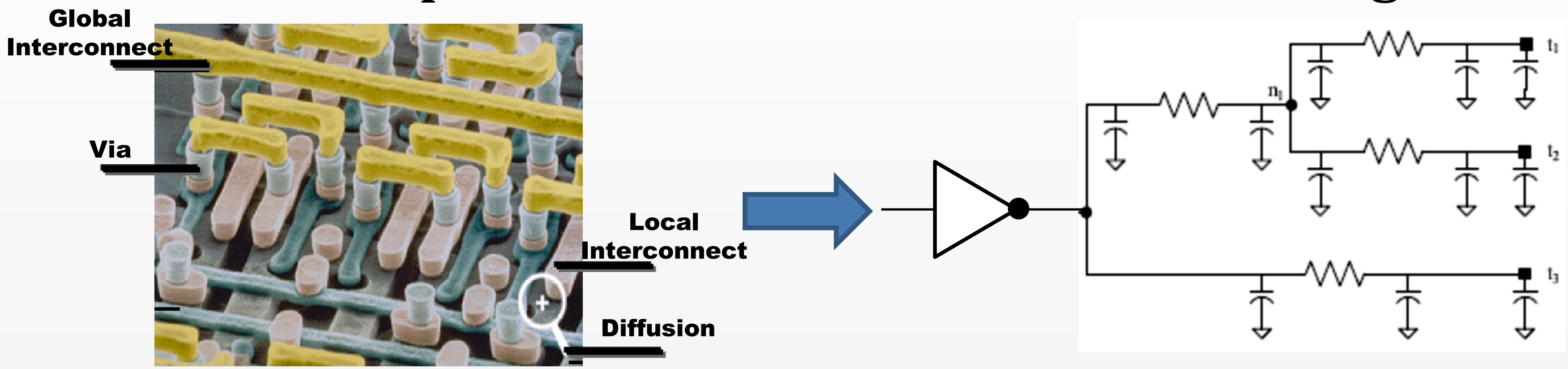
$$Q_j = \oint_{G_j} F(\boldsymbol{r}) g \oint_{S^{(1)}} \omega(\boldsymbol{r}, \boldsymbol{r}^{(1)}) q(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \phi(\boldsymbol{r}^{(1)}) \mathrm{d}s^{(1)} \mathrm{d}s$$

  - $F(\boldsymbol{r})g$ and $q(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ are PDF; $\omega(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ is called weight value
  - Sampling on G_j, $S^{(1)}$, $S^{(2)}$,…, until reaching conductor k
  - The weight value is an estimate for C_{jk}
  - Averaging the weight values after N walks produces $\{C_{ji}\}_{i=1}^{n}$

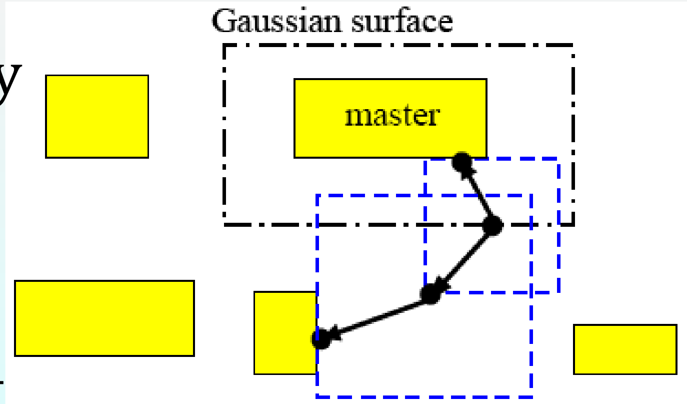
[1] **Y. Le Coz** and **R. Iverson**, "A stochastic algorithm for high speed capacitance extraction in integrated circuits," ***Solid State Electron.***, 1992
[2] **W. Yu**, et al., "RWCap: a floating random walk solver for 3-D capacitance … ," ***IEEE Trans. CAD***, 2013

MC Based Capacitance Calculation

- FRW based capacitance extraction for VLSI design



  - Calculating wire capacitances is the base of modeling & simulation
  - The secrets of the fast FRW solver
    - Cubic transition domain fits geometry
    - Numerically pre-calculate transition probabilities and weight values
    - Importance sampling; placement of Gaussian surface; space management

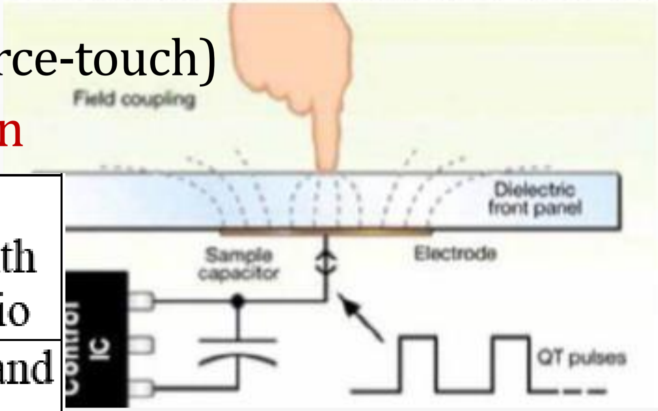


[1] **C. Zhang** and **W. Yu**, "Efficient space management techniques for large-scale ...," ***IEEE Trans. CAD***, 2013
[2] **C. Zhang**, **W. Yu**, et al., "…for the 3-D IC structures with cylindrical inter-tier-vias," ***IEEE Trans. CAD***, 2015

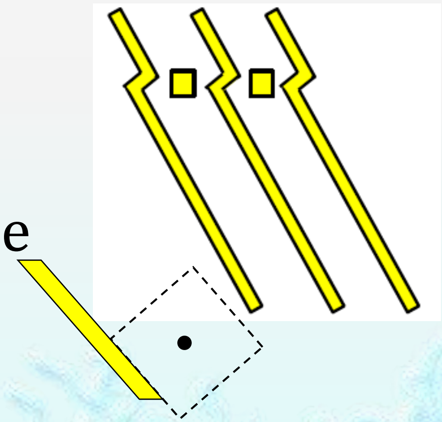
MC Based Capacitance Calculation

- FRW based capacitance simulation for touchscreen
  - Validation of functionality (multi-touch, force-touch)

C. Extraction for VLSI vs. C. Simulation for touchscreen



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

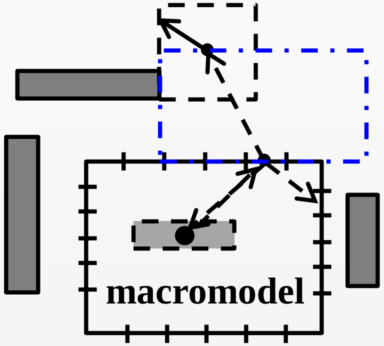


  - Geometry engine for non-Manhattan metal shape
  - Allow planar rotation of transition cube
  - A unified dielectric pre-characterization scheme
  - MPI based parallel computing on a cluster (93X~114X w/ 120 cores)

[1] **Z. Xu**, **C. Zhang**, **W. Yu**, "Floating random … non-Manhattan conductor structures," ***IEEE Trans. CAD***, 2016
[2] **Z. Xu**, **W. Yu**, et al., "A parallel random walk solver for the capacitance calculation problem in touchscreen design," ***Proc. GLSVLSI***, 2016

MC Based Capacitance Calculation

- Other progress of the FRW method
  - GPU based parallel algorithm (operation divergence/memory bottleneck)
  - Macromodel based random walk algorithm [3]: Circuits with IP protected or repeated substructures



- Related topics
  - MC method is the golden, and the sole choice for variation-aware simulation with a lot of independent variables
  - Open problem: MC (random walk) based impedance extraction
    - MC for 1-D telegraph equation
    - Applying MC to a general wave equation is still difficult

$$\frac{\partial^2 u}{\partial t^2} + \frac{R}{L}\frac{\partial u}{\partial t} = \frac{1}{LC}\frac{\partial^2 u}{\partial x^2}$$

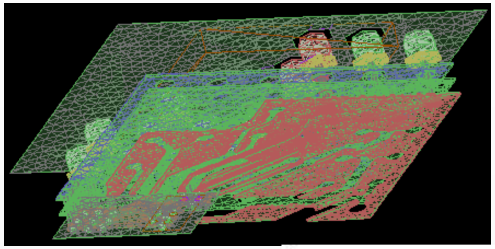
[1] **N. Arora**, **S. Worley**, and **D. Ganpule**, “FieldRC, a GPU accelerated interconnect …,” ***Proc. EDSSC***, 2015
[2] **K. Zhai**, **W. Yu** and **H. Zhuang**, “GPU-Friendly floating random walk algorithm …,” ***Proc. DATE***, 2013
[3] **W. Yu**, et al. “Utilizing macromodels in floating random walk based capacitance …,” ***Proc. DATE***, 2016
[4] **W. Yu**, et al. “Efficient statistical capacitance extraction of nanometer …," ***Microelectronics Reliability***, 2012
[5] **J. Acebron**, **M. Ribeiro**, “A Monte Carlo method for solving the one-dimensional …,” ***J. Comp. Phys.***, 2016

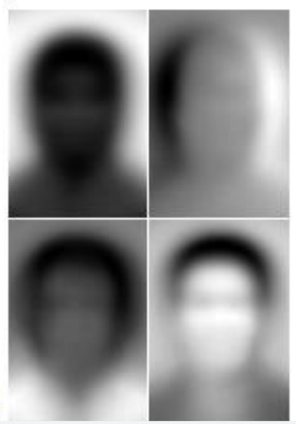
Outline

- Introduction
- Basics of Monte Carlo method
- MC based capacitance calculation
- MC based large matrix approximation
- Conclusion

MC Based Large Matrix Approx.

- Motivation
  - Large-scale EM simulation
  - "Big-Data" analytics
    - Reduced-rank LS, principal component regression
  - Artificial intelligence
    - Feature extraction, recommendation system
  - Low-rank matrix approximation/factorization plays a crucial role
    - EM solver: low-rank matrix compression; MOR: utilizes SVD
    - Principal component analysis (i.e. truncated SVD)
  - Traditional matrix decompositions have $O(n^3)$ complexity, not parallelized efficiently, need substantial visits of matrix

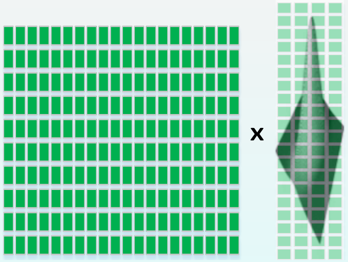


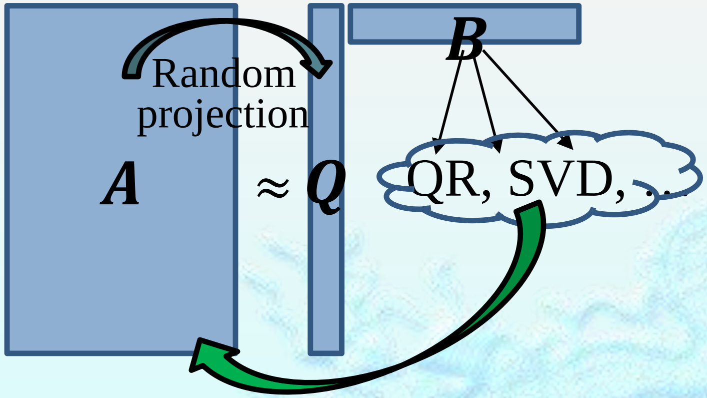


[1] **J. Phillips** and **L. Silveira**, "Poor man's TBR: a simple model reduction scheme," ***IEEE Trans. CAD***, 2005
[2] **W. Yu** and **X. Wang**, ***Advanced Field-Solver Techniques for RC Extraction of Integrated Circuits***, 2014
[3] **P. Drineas** and **M. W. Mahoney**, "RandNLA: randomized numerical linear algebra," ***Communications of the ACM***, 2016

MC Based Large Matrix Approx.

- Observations
  - Very high accuracy is not required (speed is major concern)
  - Accessing matrix entries becomes a new bottleneck
- MC based algorithms
  - Use randomization to overcome the bottleneck
  - Random column/row selection of $\boldsymbol{A}$ (week performance guarantee)
  - Random projection: randQB
    - $\boldsymbol{Q} = \mathrm{orth}(\boldsymbol{A\Omega})$
    - $\boldsymbol{B} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}$
    - Decompose short-fat $\boldsymbol{B}$, get a low-rank factorization of $\boldsymbol{A}$

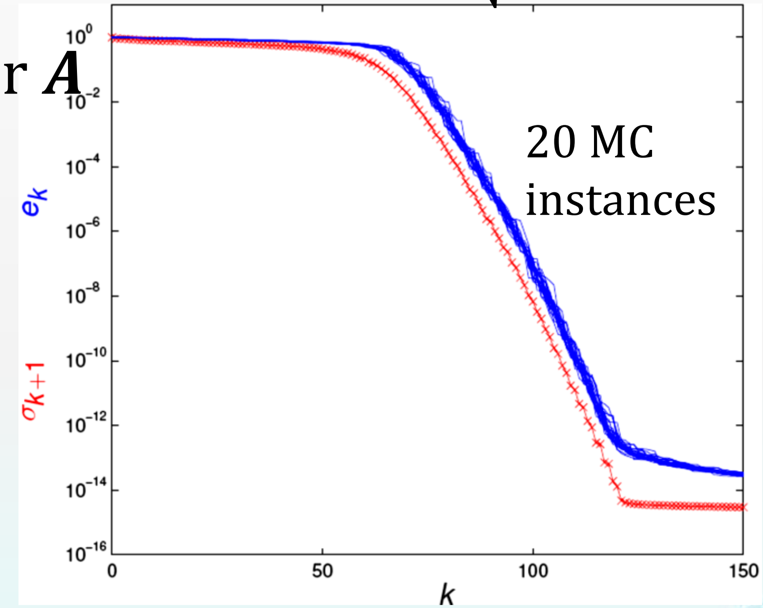




[1] **P. Drineas**, **R. Kannan**, **M. Mahoney**, "Fast Monte Carlo algorithms for matrices …," ***SIAM J. Comput.***, 2006
[2] **N. Halko**, **P.-G. Martinsson**, **J. Tropp**, "Finding structure with randomness: …," ***SIAM Review***, 2011
[3] **P. Drineas** and **M. W. Mahoney**, "RandNLA: randomized numerical linear algebra," ***Communications of the ACM***, 2016

MC Based Large Matrix Approx.

- The randQB algorithm
  - Expectation of error: $\mathbb{E}\|\boldsymbol{A}-\boldsymbol{QB}\|_2 \le \left(1+\sqrt{\frac{k}{s-1}}\right)\sigma_{k+1} + \frac{e\sqrt{k+s}}{s}\sqrt{\sum_{j=k+1}^{\min(m,n)}\sigma_j^2}$
  - Tightly concentrated:
  - Faster than QRCP; fewer passes over $\boldsymbol{A}$



- Our improvements
  - Auto-determine the rank such that

$$\|\boldsymbol{A}-\boldsymbol{QB}\| < \varepsilon$$

  - Reduce the passes over $\boldsymbol{A}$
  - Both are adaptive to large, sparse $\boldsymbol{A}$
  - Extend the algorithm for approximate matrix multiplication
  - Useful for model compression, information retrieval, etc.

[1] **N. Halko**, **P.-G. Martinsson**, **J. Tropp**, "Finding structure with randomness: …," ***SIAM Review***, 2011
[2] **Y. Gu**, **W. Yu**, **Y. Li**, "Efficient randomized algorithms for adaptive low-rank …," ***arXiv:1606.09402***, 2016

MC Based Large Matrix Approx.

- Automatic rank determination
  - A scenic picture (3168x4752 pix)
  - 10% relative error
  - ~7X data size reduction
  - ~12X faster than SVD

The rank determined

		Ours	SVD	Finder[1]
image	P=1	468	426	2913
	P=2	441		

  - A keyword-expert matrix **(8.3Kx100K)** from *AMiner.org* (TF-IDF model)
  - Also obtain near-optimal rank
  - With 10X reduction of runtime

[1] **N. Halko**, **P.-G. Martinsson**, **J. Tropp**, "Finding structure with randomness: …," ***SIAM Review***, 2011
[2] **Y. Gu**, **W. Yu**, **Y. Li**, "Efficient randomized algorithms for adaptive low-rank …," ***arXiv:1606.09402***, 2016

Outline

- Introduction
- Basics of Monte Carlo method
- MC based capacitance calculation
- MC based large matrix approximation
- Conclusion

Conclusion

- MC for deterministic quantities
  - Adapts to parallel computing, can be easily implemented
  - Much faster speed for large-scale computation, without or with less accuracy sacrifice
- Simulation with solving PDE
  - The random walk method beats deterministic methods for large-scale capacitance simulation
  - General EM simulation (wave equ.), still an open problem
- Linear algebra computation
  - Randomized technique shows power for low-rank matrix approximation
  - Lots of applications and bright perspective

Thank You !

Wenjian Yu / Tsinghua University, China

http://numbda.cs.tsinghua.edu.cn