IE 525 - Numerical Methods in Finance

Monte Carlo simulation - Quasi-Monte Carlo

Liming Feng

Dept. of Industrial & Enterprise Systems Engineering
University of Illinois at Urbana-Champaign

Variance reduction vs QMC

- We want to estimate $\mathbb{E}[Y]$
- Standard error in a direct approach with sample size n ($\sigma^2 = \text{var}(Y)$)

$$\frac{\sigma}{\sqrt{n}}$$

- **Variance reduction** techniques reduce the variance σ^2 to improve efficiency
- **Quasi-Monte Carlo** improves efficiency by speeding up the convergence from $1/\sqrt{n}$ to almost $1/n$

Dimension

- Suppose Y can be generated from independent uniform random variables: $Y = f(U_1, \cdots, U_d)$

$$\mathbb{E}[Y] = \mathbb{E}[f(U_1, \cdots, U_d)] = \int_{[0,1)^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

  for carefully chosen points $x_1, \cdots, x_n \in [0,1)^d$
- For quasi-Monte Carlo, the **dimension d matters**! The smaller the dimension, the better; for infinite d (e.g., acceptance-rejection method), it doesn't work

Low discrepancy

- Select $x_1, \cdots, x_n$ so that they fill $[0,1)^d$ as uniformly as possible (**low discrepancy**)
- For a collection $\mathcal{A}$ of subsets of $[0,1)^d$ and the set $\{x_1, \cdots, x_n\}$, **discrepancy** is defined to be

$$D(x_1, \cdots, x_n; \mathcal{A}) = \sup_{A \in \mathcal{A}} \left| \frac{\#\{x_i \in A\}}{n} - vol(A) \right|$$

$$\frac{\#\{x_i \in A\}}{n} - vol(A) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_A(x_i) - \int_{[0,1)^d} \mathbf{1}_A(x) dx$$

- Discrepancy is maximal approximation error for indicator functions $\mathbf{1}_A(x)$

Ordinary and star discrepancies

- Discrepancy depends on $\mathcal{A}$
- **Ordinary discrepancy** $D(x_1, \cdots, x_n)$: $\mathcal{A}$ is collection of rectangles of the form

$$\Pi_{j=1}^{d}[u_j, v_j),\ 0 \leq u_j < v_j \leq 1$$

- **Star discrepancy** $D^*(x_1, \cdots, x_n)$: $\mathcal{A}$ is collection of rectangles of the form

$$\Pi_{j=1}^{d}[0, u_j)$$

Example: 1-d

- When $d = 1$, both ordinary and star discrepancies are minimized with

$$x_i = \frac{2i - 1}{2n}, \ i = 1, \cdots, n$$

  The corresponding discrepancies are of the form $\frac{c}{n}$ for some $c > 0$
- When increasing n, x_i's could be totally different and need be recomputed
- Prefer a long sequence: for any n, use the first n numbers and still achieve low discrepancy

- Can find infinite sequence $\{x_1, x_2, \cdots\} \subset [0,1)$ such that the discrepancies are of the form

$$O\left(\frac{\log n}{n}\right)$$

- Use a sequence that is computationally more convenient by sacrificing a little uniformity
- More generally, for d dimensional problems, can find infinite sequence $\{x_1, x_2, \cdots\} \subset [0,1)^d$ such that the discrepancies are of the form

$$O\left(\frac{(\log n)^d}{n}\right)$$

or slightly looser bound $O\left(\frac{1}{n^{1-\epsilon}}\right)$ for arbitrary small $\epsilon > 0$

Koksma-Hlawka

- **Koksma-Hlawka bound** for quasi-Monte Carlo estimation error

$$\left|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \int_{[0,1)^d} f(x)dx\right| \leq cD^*(x_1, \cdots, x_n)$$

  where c depends on f and d but not on n or $\{x_1, \cdots, x_n\}$
- Quasi-Monte Carlo estimation error can be made of the following form by using low discrepancy sequences

$$O\left(\frac{(\log n)^d}{n}\right) \text{ or loosely } O\left(\frac{1}{n^{1-\epsilon}}\right)$$

MC vs QMC

- In each case, we approximate $\mathbb{E}[f(U_1, \cdots, U_d)]$ by $\frac{1}{n}\sum_{i=1}^{n} f(x_i)$

MC	QMC
simulate $x_i = (u_1^i, \cdots, u_d^i)$, where u_k^i is from $U[0,1]$. We make x_i's random	determine x_i's strategically, x_i's are non-random
estimation error bounded by $z_{\alpha/2}\sigma/\sqrt{n}$ with probability $1-\alpha$	estimation error bounded by $c(\log n)^d/n$, strictly
the above bound computable	difficult to compute c above
performance doesn't depend on d	worse performance for larger d
assume finite $\sigma^2 = \text{var}(f(U_1, \cdots, U_d))$	assume finite variation for f

Case of dimension 1

- Given an integer $b \geq 2$ (base), any integer $k > 0$ admits a unique representation

$$k = \sum_{j=0}^{\infty} a_j(k) b^j, \ 0 \leq a_j(k) < b$$

- Define $\psi_b(k) = \sum_{j=0}^{\infty} \frac{a_j(k)}{b^{j+1}}$
- Example: with base $b = 10$,

$$k = 1234 = 10^0 \times 4 + 10^1 \times 3 + 10^2 \times 2 + 10^3 \times 1, \psi_b(k) = 0.4321$$

Base 2

- Example: with base $b = 2$,

k	a_j's	$\psi_b(k)$	k	$\psi_b(k)$
1	1	1/2	8	1/16
2	10	1/4	9	9/16
3	11	3/4	10	5/16
4	100	1/8	11	13/16
5	101	5/8	12	3/16
6	110	3/8	13	11/16
7	111	7/8	14	7/16

- The corresponding base 2 sequence is $\{\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \cdots\}$
- Such sequences are called **Van der Corput sequences**; they are low discrepancy for any $b \geq 2$

Extensions to dimension d

- **Halton** sequence in $[0,1)^d$ is obtained by using a Van der Corput sequence with base b_i along ith dimension, where $b_1, \cdots, b_d$ are relatively prime (performance deteriorates fast as d increases)
- **Faure** sequence in $[0,1)^d$ is obtained by using a permuted Van der Corput sequence with prime base $b \geq d$ along each dimension (base becomes large as d increases)
- **Sobol'** sequence in $[0,1)^d$ is obtained by using a permuted Van der Corput sequence with base 2 along each dimension

Randomized QMC

- Quasi-Monte Carlo gives estimates with strictly bounded error; but the exact bound is difficult to compute
- Monte Carlo provides confidence intervals, but converges slowly
- **Randomization** of quasi-Monte Carlo combines (1) better accuracy of quasi-Monte Carlo (2) easiness for computing confidence intervals of Monte Carlo

Random shift

- Denote $P_n = \{x_1, \cdots, x_n\}$, a low discrepancy sequence of size n
- **Random shift**: simulate a random vector $U = (U_1, \cdots, U_d)$, where $U_1, \cdots, U_d$ are i.i.d and uniform on $(0, 1)$; define

$$P_n(U) = \{x_i + U \bmod 1, 1 \leq i \leq n\}$$

- $x \bmod 1 = x - floor(x)$: e.g., $1.3 \bmod 1 = 0.3$; $0.8 \bmod 1 = 0.8$
- The kth entry of each x_i is shifted by $U_k \bmod 1$
- Randomized quasi-Monte Carlo estimate

$$I_f(U) = \frac{1}{n} \sum_{i=1}^{n} f(x_i + U \bmod 1)$$

Confidence interval from random shift

- Any $x_i + U$ mod 1 is uniform on $[0,1)^d$. $I_f(U)$ is thus unbiased estimate of $I_f = \int_{[0,1)^d} f(x)dx$
- Repeating the above randomization for L times (e.g., $L = 10$ or 20) generates L i.i.d. estimates $I_f(U^1), \cdots, I_f(U^L)$
- Report the average of the L estimates
- Confidence intervals can be constructed using the L estimates
- f is evaluated for a total of nL times. Compare to Monte Carlo with sample size nL.
- Efficiency measure:

$$(\textit{standard error})^2 \times \textit{total computational time}$$

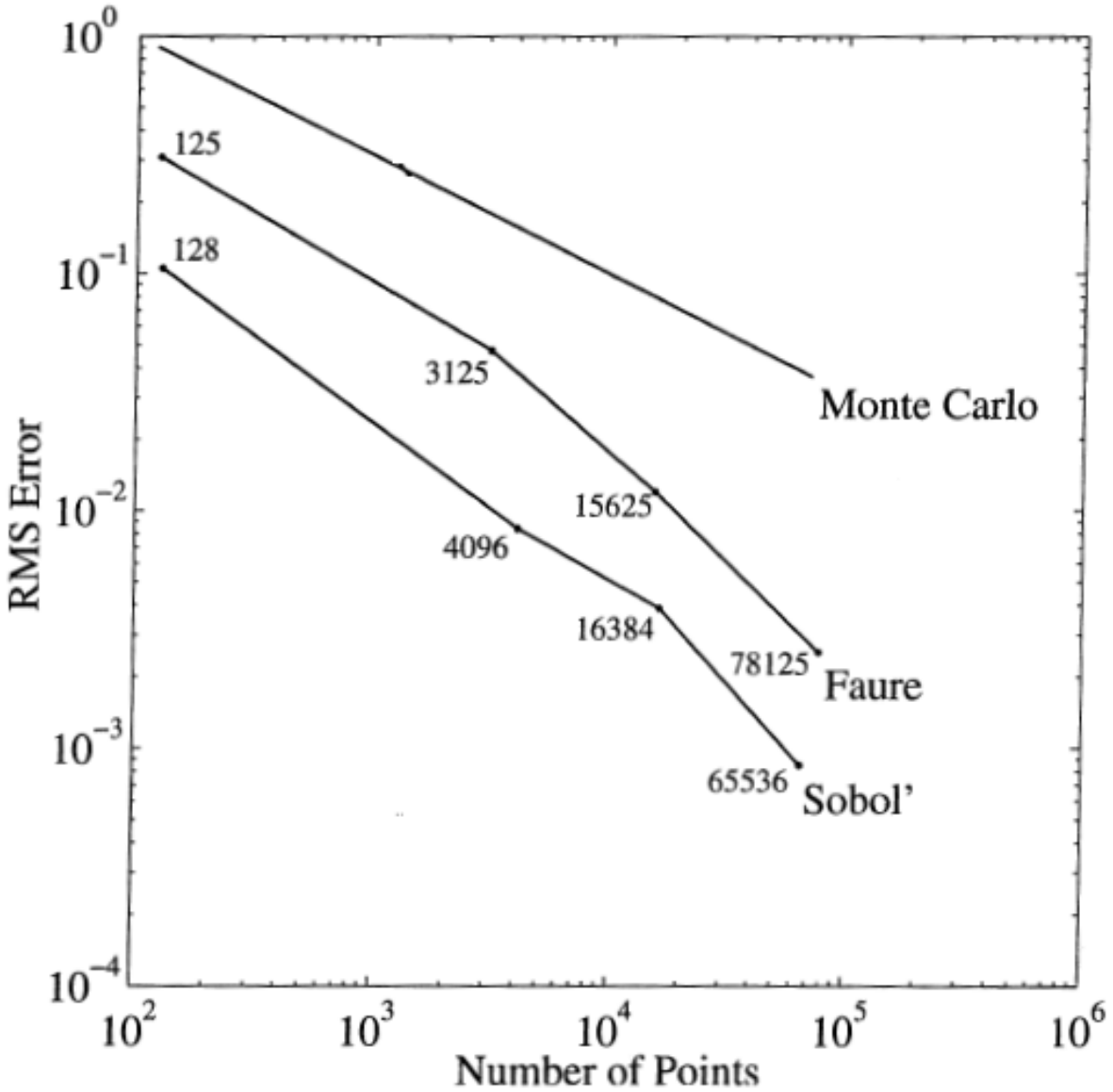
Example: multi-asset options

- Consider a call option with maturity T, strike price K and payoff
$$(\bar{S} - K)^+, \quad \bar{S} = \Pi_{i=1}^{d} S_i(T)^{1/d},$$
where $S_1(T), \cdots, S_d(T)$ are the prices of d stocks at maturity
- Assume BSM model for the stock prices. $\bar{S}$ is lognormal. The option price admits closed-form expressions similar to Black-Scholes formula
- When $\bar{S}$ is arithmetic average, it's known as a **basket option**. No closed form solution
- Use geometric average with closed form solution to evaluate effectiveness of quasi-Monte Carlo

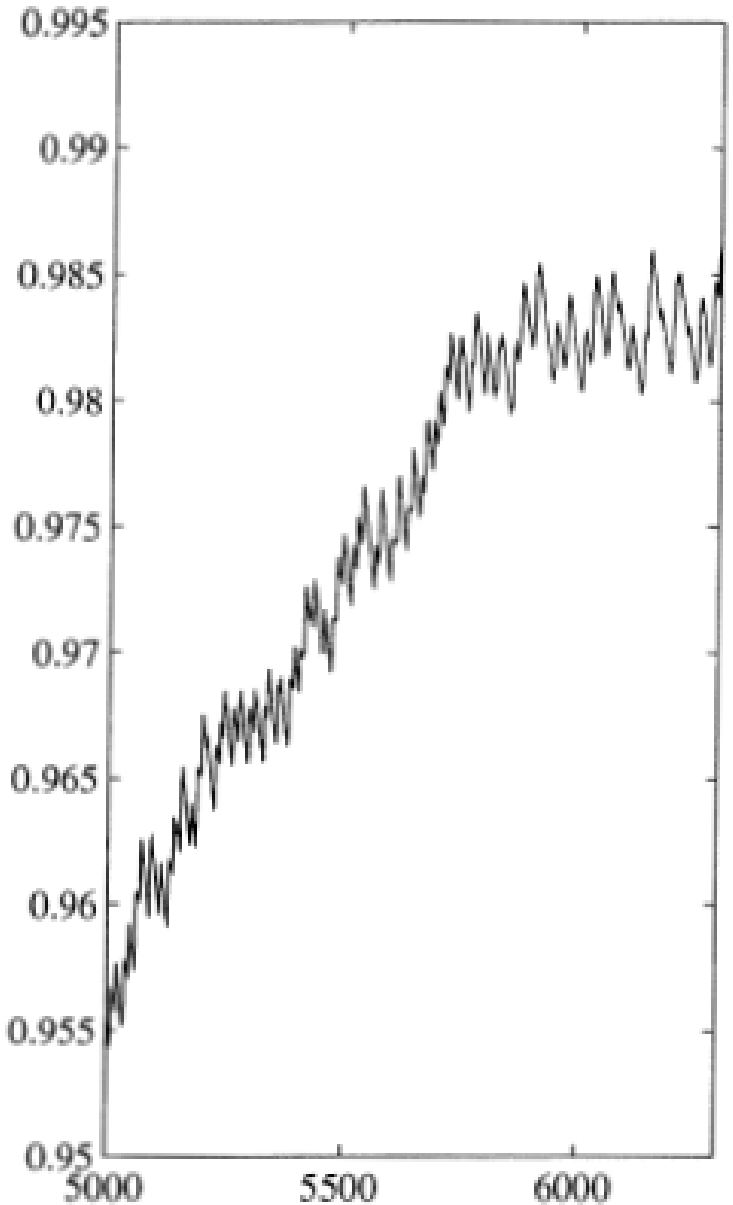
- Suppose the stocks are independent. Consider 500 options with: $T = 0.15, 0.25, 0.5, 1, 2$, $\sigma = 0.21 : 0.05 : 0.66$, $K = 94 : 1 : 103$, $S_i(0) = 100, i = 1, \cdots, 5$, $r = 5\%$
- The average pricing error below compares average performance of various different methods

$$RMSE(n) = \Big(\frac{1}{500}\sum_{k=1}^{500}(\hat{C}_k(n) - C_k)^2\Big)^{1/2}$$

C_k is the true price of kth option. $\hat{C}_k(n)$ is the estimate of kth option price using sample of size n


RMS Error
Number of Points
Monte Carlo
Faure
Sobol'
125
128
3125
15625
4096
16384
78125
65536

- Sobol' sequence outperforms Faure sequence in the example; both outperform Monte Carlo
- Slope of Monte Carlo $-1/2$; slopes of quasi-Monte Carlo methods close to -1
- If more points were added, the quasi-Monte Carlo graphs may appear more erratic. But the trend remains the same

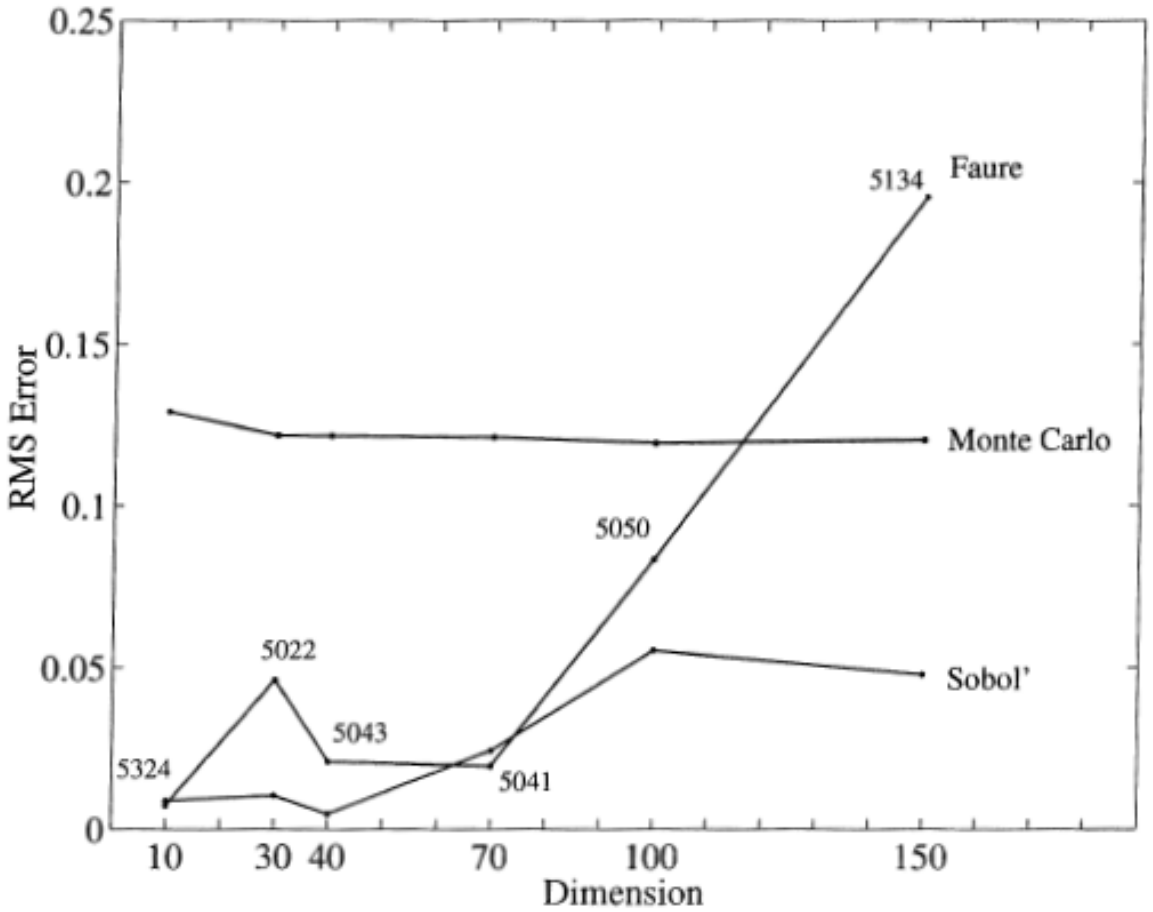

0.995
0.99
0.985
0.98
0.975
0.97
0.965
0.96
0.955
0.95
5000
5500
6000

Example: geometric Asian call

- Consider a geometric Asian call with maturity T, strike price K and payoff

$$(\bar{S} - K)^+, \quad \bar{S} = \Pi_{i=1}^{d} S_{i\delta}^{1/d}, \quad \delta = T/d$$

  where $S_{i\delta}$'s are the prices of a stock at times $i\delta, i = 1, \cdots, d$
- Closed form solution for the option price available in the BSM
- Consider the average pricing error of 50 options with $T = 0.25, K = 96 : 2 : 104, \sigma = 0.21 : 0.05 : 0.66$, $S_0 = 100$, $r = 5\%$


0.25
0.2
0.15
0.1
0.05
0
RMS Error
10
30
40
70
100
150
Dimension
Faure
Monte Carlo
Sobol'
5134
5050
5022
5043
5041
5324

- Performance of Monte Carlo remains nearly constant across dimensions
- Quasi-Monte Carlo generally gets worse when dimension increases
- Sobol' sequence remains competitive for high dimensions in this example

Final remarks

- Monte Carlo is slow convergent; but provides measure of precision; can be more efficient when combined with appropriate variance reduction techniques
- To determine necessary sample size for desired precision, first do a pilot run (with several hundred or thousand replicates) to determine variance
- Quasi-Monte Carlo converges faster; but difficult to know what sample size is sufficient
- Sobol' sequence often performs better in financial applications
- Randomization helps produce measure of precision; can be combined with variance reduction techniques as well