Risk Management

Monte Carlo Simulation Techniques

Monte-Carlo Simulation Techniques

- Pseudo-Random Number Generation
- Generating Pseudo-Random Variables
- Forecasting Volatilities and Correlations
- Monte Carlo Simulation of Diffusions
- Monte Carlo Options Pricing
- Variation Reduction Techniques

Modeling Financial Processes

- Geometric Brownian Motion

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t \qquad S_t = S_0 \exp\left(\left[\mu - \frac{1}{2}\sigma^2\right]t + \sigma W_t\right)$$

- Hull White Stochastic Volatility Model

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t^1 \qquad \frac{d\sigma_t^2}{\sigma_t^2} = \nu dt + \xi dW_t^2$$

One Factor Interest Rate Models

- General Form: $dr = m(r)\,dt + \sigma(r)\,dW$
  - Ito Process:
    - m: drift factor
    - σ: short rate volatility
    - dw: $\varepsilon\sqrt{t}$; $\varepsilon \sim N(0,1)$
- Model characteristics
  - All rates move in same direction, but not by same amount
  - Many different shapes possible (including inverted)
  - Mean reversion can be built in

Model Taxonomy

	Expected change in r m(r)	Mean Reversion	Volatility of r s(r)	Fits Term Str. Yield	Fits Term Str. Vol
Vasicek	a[m - r]	yes	constant	no	no
CIR	a[m - r]	yes	$\sigma\sqrt{r}$	no	no
Brennan & Schwartz	a[b + L - r]	yes	f(r,L)	no	no
Ho & Lee	g(t)	no	constant	yes	no
BDT	f(t,r,σ)	limited	f(time)	yes	yes
Hull & White	a(t)[m(t) - r]	yes	f(time)	yes	yes

Pseudo-Random Number Generation

- Simulation of price and return paths over time
  - Requires the generation of sequences of random variables.
  - This is known as Monte Carlo sampling
  - Generation of "random" (i.e. almost independent) numbers
- Pseudo-Random Numbers
  - Represent as decimal fractions
  - Interpret as realizations U of the uniform distribution on the unit interval U(0,1)

Linear Congruential Generator

- Most Common Method of Sequential Generation

  $x_{i+1} = (ax_i + c)$ modulo m   i = 0, 1, 2, ...

  - m is a very large number -- the period of the generator
  - a and c are parameters
  - $x_0 \in \{0, 1, \ldots, m-1, m\}$

  is the seed provided to start the recursive stream of numbers x_0, x_1, x_2, ...

- Construct uniform pseudo-random variates $\mathbf{u}_i \sim U(0,1)$

  $u_i = x_i / m \quad i = 0, 1, 2, \ldots$

  - Sequence is not independent due to the m-long cycle
  - Okay when the sample number n is small relative to m

Example

For example, if $x_0 := 35$, $a := 13$, $c := 65$, and $m := 100$ the algorithm works as follows:

Iteration 0 Set $x_0 = 35$, $a = 13$, $c = 65$, and $m = 100$.

Iteration 1 Compute

$$x_1 = (a\, x_0 + c) \text{ modulo } m$$
$$= [13(35) + 65] \text{ modulo } 100$$
$$= 20$$

Deliver

$$u_1 = x_1 / m = 20/100 = 0.2$$

Example (continued)

Iteration 2 Compute

$$x_2 = (a\ x_1 + c) \text{ modulo } m$$
$$= [13(20) + 65] \text{ modulo } 100$$
$$= 25$$

Deliver

$$u_2 = x_2 / m = 25/100 = 0.25$$

Iteration 3 Compute

$$x_3 = (a\ x_2 + c) \text{ modulo } m$$

.

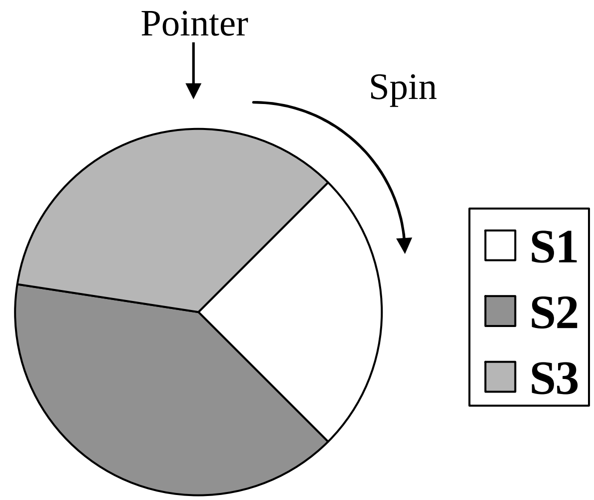
.

and so on.

Generating Discrete Random Variables

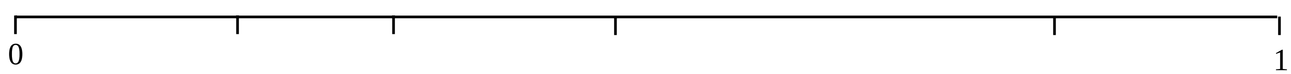
- Conceptually we may generate all n-valued discrete random variables according to the following scheme for S taking 3 values with p.d.f.:

$$f_s(s) := \begin{cases} 0.25 & s = S_1 \\ 0.40 & s = S_2 \\ 0.35 & s = S_3 \\ 0 & O.W. \end{cases}$$



Discrete PDF Generation

- More generally for **x** with discrete p.d.f:
  - We divide the unit interval as



since $$\sum_{n=1}^{N} f_{\mathbf{x}}(x_n) = 1 \quad n = 1, \ldots, N$$

and return x_n iff u_n falls in the n^{th} interval

Example: Bootstrapping Daily Returns

- Consider the problem of simulating "Monthly" returns **x** over N trading days per "Month" given a sequence x_1, ..., x_N of actual **daily returns for one such month**

- The empirical density function for the return process **x** assigns probability 1/N to each actual observation -- assuming the data independently and identically distributed -- and we think of $\#\{x_i < k\}$ as the empirical estimate of $P\{\mathbf{x} < k\}$

Resampling/Bootstrapping

- Now we use the uniform pseudo-random numbers u_0, u_1, u_2, ... to generate x_n n = 1, ..., N iff:

$$u_i \in \left[\frac{n-1}{N}, \frac{n}{N} \right]$$

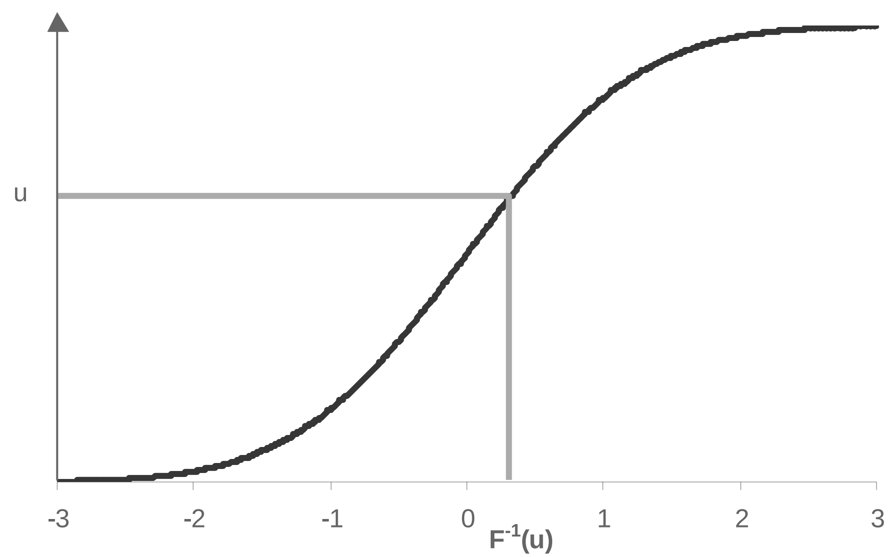
for as many multiples of N as is required

- This resampling technique -- known as bootstrapping -- may be used to generate as many monthly returns as are required as e.g.:

$$\sum_{i=q+1}^{q+N} x_i \qquad \text{for month q} = 1, 2, \ldots$$

Generating Continuous Random Variables

- Inverse Transform Method
  - Used for random variables **x** whose c.d.f. is available in closed form $F_{\mathbf{x}}$
  - Based on $P\{\mathbf{U} \le u\} = P\{\mathbf{x} \le F_{\mathbf{x}}^{-1}(u)\}$



Generating Normal Variates

- In finance we are most interested in generating standard normal random variables **z** ~ $N(0,1)$
- Generate **x** ~ $N(\mu, \sigma^2)$ as $\mathbf{x} = \mu + \sigma\,\mathbf{z}$
  - Noting that **u** ~ U(0,1) has mean 1/2 and variance 1/12 the central limit theorem means that

$$\mathbf{z} = \left(\sum_{i=1}^{n} \mathbf{u}_i - n/2\right) \Big/ (n/12)^{1/2} \approx \mathrm{N}(0,1)$$

  - In practice n := 12 to give $\mathbf{z} = \sum_{i=1}^{12} \mathbf{u}_i - 6$

Box-Muller Algorithm (1958)

- More accurate is the exact Box-Muller (1958) transformation of $\mathbf{u}_1, \mathbf{u}_2 \sim U(0,1)$ independent r.v's as

$$\mathbf{z}_1 = (-2\ln\mathbf{u}_1)^{1/2}\sin 2\pi\mathbf{u}_2 \sim N(0,1)$$

$$\mathbf{z}_2 = (-2\ln\mathbf{u}_1)^{1/2}\cos 2\pi\mathbf{u}_2 \sim N(0,1)$$

Generating Random Vectors

- Need to Generate Random Vectors
  - For Correlated Returns
    - i.e. Log of price ratios
    - Assumed multivariate Normal
  - For Multivariate (Log-Normal) Prices
    - Used for risk management & stress testing

Correlated Normal Variates

- Instantaneous correlation ρ
- Generate correlated bivariate Normal variates
  - From standard normal variates ε_1 and ε_2

$$w_1 = \varepsilon_1$$

$$w_2 = \rho\varepsilon_1 + \sqrt{1-\rho^2}\,\varepsilon_2$$

Procedure for Generating Random Vectors

- Given (estimated) covariance matrix Σ
  - Factor the covariance matrix Σ into **Cholesky factors** as $\Sigma = A\,A'$
  - Generate z ~ $N(0,1)$
  - Generate returns $\mathbf{x} = A\mathbf{z} \sim N(0, \Sigma)$
  - Given a vector f of expected future spot prices generate multivariate lognormal prices **p** for the next period as $\mathbf{p}_i = f_i e^{\mathbf{x}_i} \quad \mathrm{i} = 1, \ldots, \mathrm{I}$

Cholesky Factorization

- Compute Cholesky factorization $\Sigma = A\,A'$ -- where A is lower triangular -- in $O(n^2)$ operations as follows

$$\Sigma = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} A = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} A' = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}$$

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11}^2 & a_{11}a_{21} & a_{11}a_{31} \\ a_{11} & a_{21}^2 + a_{22}^2 & a_{21}a_{31} + a_{32}a_{22} \\ a_{11}a_{31} a_{21}a_{31} + a_{32}a_{22} & a_{11}^2 + a_{22}^2 + a_{33}^2 & \end{bmatrix}$$

Cholesky Factorization

- Now we use the elements s_{ij} of Σ to solve for the elements a_{ij} of A -- positive definite
- For an IxI matrix Σ we use the recursions

$$a_{ii} = [s_{ii} - \sum_{k=1}^{i-1} a_{ik}^2]^{1/2}$$

$$a_{ij} = [s_{ij} - \sum_{k=1}^{i-1} a_{ik} a_{jk}] / a_{ii} \quad \mathrm{i} = 1, ..., \mathrm{I}, \;\; \mathrm{j} = \mathrm{i} + 1, \mathrm{i} + 2, ..., \mathrm{I}$$

Monte Carlo Simulation of Diffusions

- Return or price process **S** given by a stochastic differential equation of the form

$$\frac{dS_t}{S_t} = \mu(S_t, t)dt + \sigma(S_t, t)d\mathbf{W}_t$$

  - We may apply the theory previously developed in both the univariate and multivariate cases
  - In the univariate case we consider a real time increment Δt and simulate a realization of the state $\mathbf{S}_t$ of the process for t=0, 1, 2 ... where t = kΔt

Simulating a Univariate Diffusion Process

- Wiener Process **W** is given by

$$\Delta \mathbf{W}_t = \sqrt{\Delta t}\,\mathbf{z}_t \, \mathrm{t} = 0, 1, 2, ...$$

  - Where $\{\mathbf{z_t}\}$ are i.i.d. $N(0,1)$

- So

$$\mathbf{W}_t = \sqrt{\Delta t}\sum_{s=0}^{t-1} \mathbf{z}_t \quad \mathrm{t} = 0, 1, 2, ...$$

Procedure for Univariate Process

- Generate stream of pseudo-random standard normal variates z_0, z_1, z_2, ...
- Obtain stream of pseudo-random Wiener process increments ΔW_0, ΔW_1, ΔW_2, ...
- Produces stream of diffusion process increments ΔS_0, ΔS_1, ΔS_2, ... which are added to give the current state realization as …

$$S_t = \sum_{i=0}^{t-1} \Delta S_i + S_0$$

Multiple Stochastic Factors

- Example: Spread Option
  - Difference between two assets

$$\frac{dS_{1t}}{S_{1t}} = \mu_1 dt + \sigma_1 dW_{1t} \quad \frac{dS_{2t}}{S_{2t}} = \mu_2 dt + \sigma_2 dW_{2t}$$

- Hull White Stochastic Volatility Model

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t^1 \quad \frac{d\sigma_t^2}{\sigma_t^2} = \nu dt + \xi dW_t^2$$

Multiple Stochastic Factors

- Example with mean reverting square root volatility process (Hull White 1988)

$$\frac{dS_{1t}}{S_{1t}} = \mu_1 dt + \sigma_1 dW_{1t} \qquad \frac{dS_{2t}}{S_{2t}} = \mu_2 dt + \sigma_2 dW_{2t}$$

$$d\sigma_{1\mathrm{t}}^2 = \alpha_1(\overline{\sigma}_{1\mathrm{t}}^2 - \sigma_{2\mathrm{t}}^2)dt + \xi_1 \sigma_{1\mathrm{t}} dW_{3\mathrm{t}}$$

$$d\sigma_{2\mathrm{t}}^2 = \alpha_2(\overline{\sigma}_{2\mathrm{t}}^2 - \sigma_{2\mathrm{t}}^2)dt + \xi_2 \sigma_{2t} dW_{4\mathrm{t}}$$

Simulating Multivariate Diffusion Processes

- Multivariate case of I instruments
  - S_t is an I-vector
- Covariance Σ and drift μ must be estimated
- Volatility σ in the diffusion S.D.E. is its **Cholesky factor**, i.e. Σ = σσ' and

$$\frac{\Delta S_t}{S_t} = \mu(S_t, t)\Delta t + \sigma(S_t, t)\Delta W_t \qquad t = 0, 1, 2, \ldots$$

Monte Carlo Option Pricing

- Simulate paths of geometric Brownian motion in terms of

$$\Delta S_t = rS_t\Delta t + \sigma S_t \Delta W_t \quad \text{t} = 0, 1, 2, \ldots$$

  - This approach must be used for exotic -- path dependent -- options such as lookbacks and Asians
- So far there are no reliable methods for simulation of **American option** prices
  - Due to the unknown exercise boundary
  - See Brodie & Glasserman 1996, Rebonato & Cooper 1996

Monte Carlo Pricing of Vanilla European Options

- Valuation under the **risk neutral measure** given by

$$P(S,t) := e^{-r(T-t)}\mathrm{E}[\, f(\mathbf{S}_t, T) \,|\, \mathbf{S}_t = S\,]$$

  - Evaluate the **conditional expectation** by Monte Carlo methods

Example: European call

- Involves the integral expectation

$$\int_X^{\infty} (e^{\ln S} - X) f_{\ln s_t}(s) ds$$

  - where $f_{\ln s_t}$ is the N\{(T-t)r,(T-t)σ^2\} density

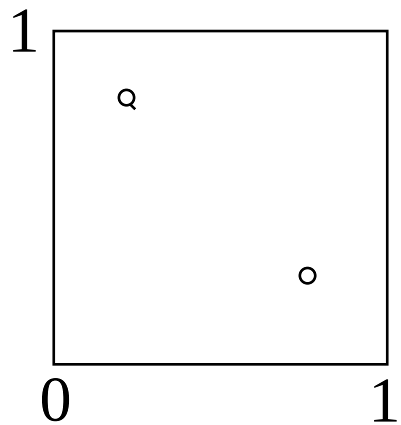
- Change variables to invert the normal c.d.f. using table lookups and interpolation
- Convert this to a new integral of an integrand g on the unit interval

$$I(g) := \int_0^1 g(\xi) d\xi \approx \frac{1}{N} \sum_{n=0}^{N-1} g(u_n) \qquad u_n \in [0,1]$$

Variance Reduction Techniques: Antithetic Variates

- These are used to **speed convergence** of the Monte Carlo approximation and the most popular are the following
- **Antithetic Variates** Use both u and 1-u to double sample size cheaply

d:=2



$$V_{est} = \frac{1}{2}\left[V_{est}\{u\} + V_{est}\{1-u\}\right]$$

- As long as covariance between V{z} and V{1-z} is negative, the overall variance will be substantially reduced

Variance Reduction Techniques: Stratification

- Random sample
  - Tends to leave gaps
- Stratified sample gives more regular representation
  - $V_i = (i-1) + U_i / 100$
    - U_i are iid uniform variates
    - Each V_i is uniformly distributed in the (i-1)th percentile
    - Then $Z_i = \Phi^{-1}(V_i)$ falls between (i-1) and ith percentile of Normal distribution
  - Downside is loss of independence
    - Critical for statistical inference

Variance Reduction Techniques: Control Variates

- **Control Variates**
  - Correct Monte Carlo estimate of exotic value with vanilla MC error

$$\hat{V}^{E} = V^{EMC} + (V^{BSMC} - V^{BS})$$

- Variate – Control Variate correlation
  - Reduces estimate variance when control and variate are correlated
    - Based on cancellation of shared estimation errors
  - No benefit if control is uncorrelated with variate,
  - If negative correlated, may increase estimate variance!

Control Variates

- Example: Barrier option
  - Price V_i = discounted payoff for path i

$$V_i = h(S_0, S_{t1}^{(i)}, \ldots\ldots\ldots\ldots, S_{tm}^{(i)})$$

  - MCS estimate V = E[V_i]

$$\frac{1}{n}\sum_{i=1} V_i$$

- Assume standard call option control variate
  - Price known in closed form, or easily evaluated

$$C = E[C_i] = E[g(S_0, S_{t1}^{(i)}, \ldots\ldots\ldots, S_{tm}^{(i)}]$$

Control Variates

- Controlled Estimator

$$\frac{1}{n}\sum_{i=1}^{n} V_i - \beta\left(\frac{1}{n}\sum_{i=1}^{n} C_i - C\right)$$

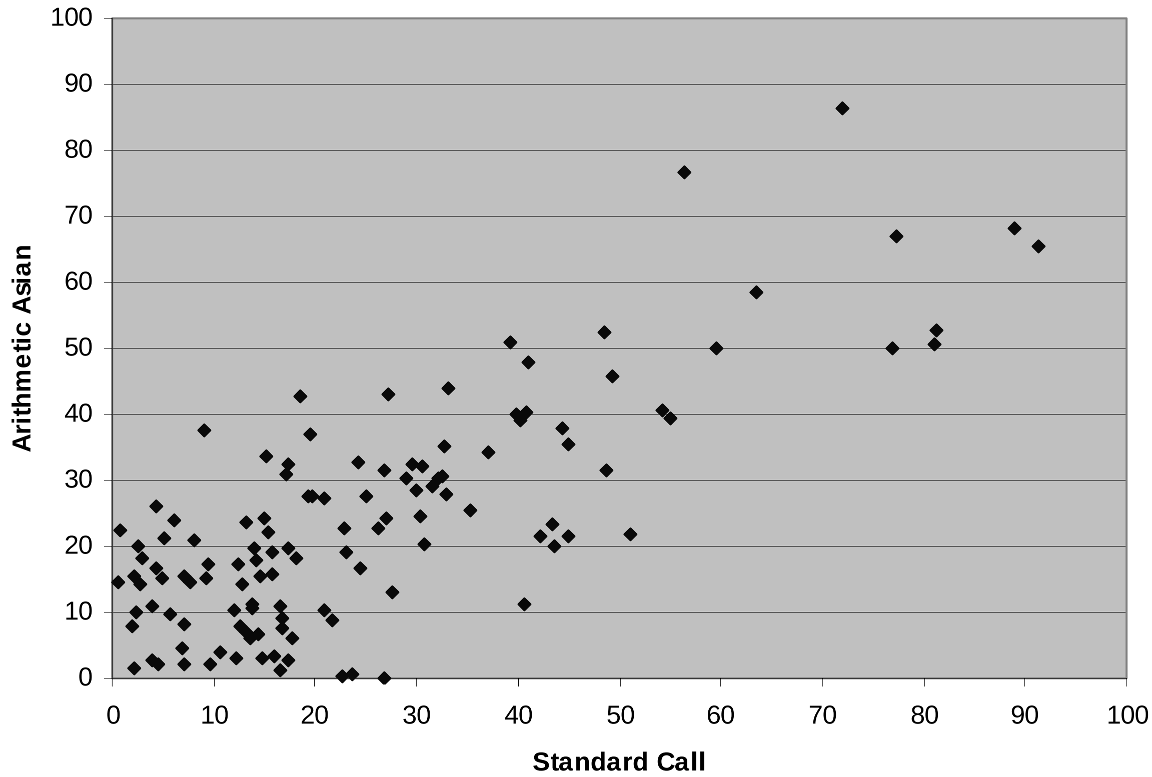
  - Coefficient with smallest variance is:

$$\beta^* = \frac{Cov[V_i, C_i]}{Var[C_i]}$$

  - Variance reduction of estimator: $1 - \rho^2_{v,c}$

Control Variate Example

- Correlation = 0.78
- Variance reduction = 61%



Quasi-Random Numbers

- Low Discrepancy Sequences
- Deterministic sequences generated by number theory
  - **Halton , Sobel, Faure**
  - Sequences appear random, but not "clumpy"
  - Behavior is ideal for fast convergence

Example: Van der Corput Sequence

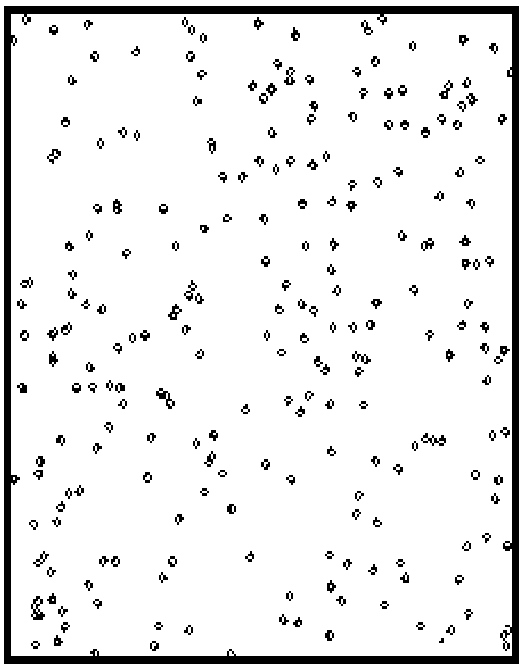
- To obtain nth point in series x_n
  - Restate n in base 2 $$n = \sum_{i=0}^{I} a_i 2^i$$
  - Transpose digits in a_i around "decimal point"
    $$x_n = \sum_{i=0}^{I} \frac{a_i}{2^{i+1}}$$
  - Generates ½, ¼, ¾, 1/8, 5/8, 3/8, 7/8
    - Contained in [0,1]
    - Every consecutive quadruple of points has one point in
      - (0, ¼), [1/4, ½), ½, ¾), [3/4, 1)

Other Sequences

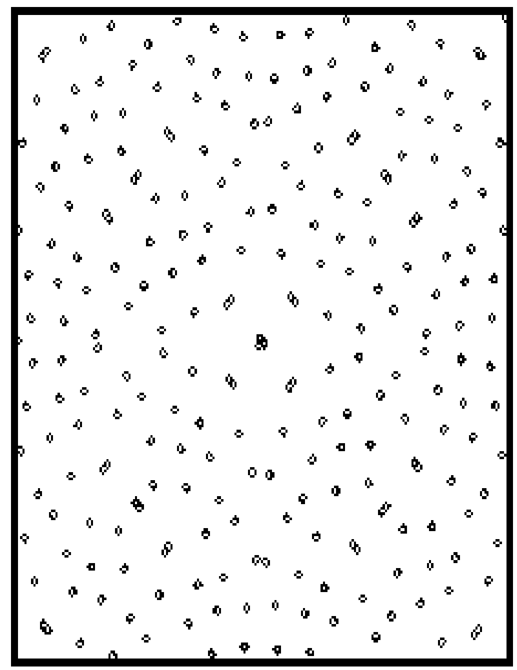
- Halton
  - General s-dimensional sequence in $[0,1]^s$ hypercube
    - First dimension is van der Corput base 2
    - Second dimension is van der Corput base 3
    - S-dimension is van der Corput base s^{th} prime number
- Faure
  - All dimensions use base prime p >= s >= 2
  - First dimension sequence is van der Corput base p
  - Higher dimensions are permutations of 1st dim.
- Sobol
  - All dimensions use 2 as base

Random vs. Sobel

Paskov (1997)

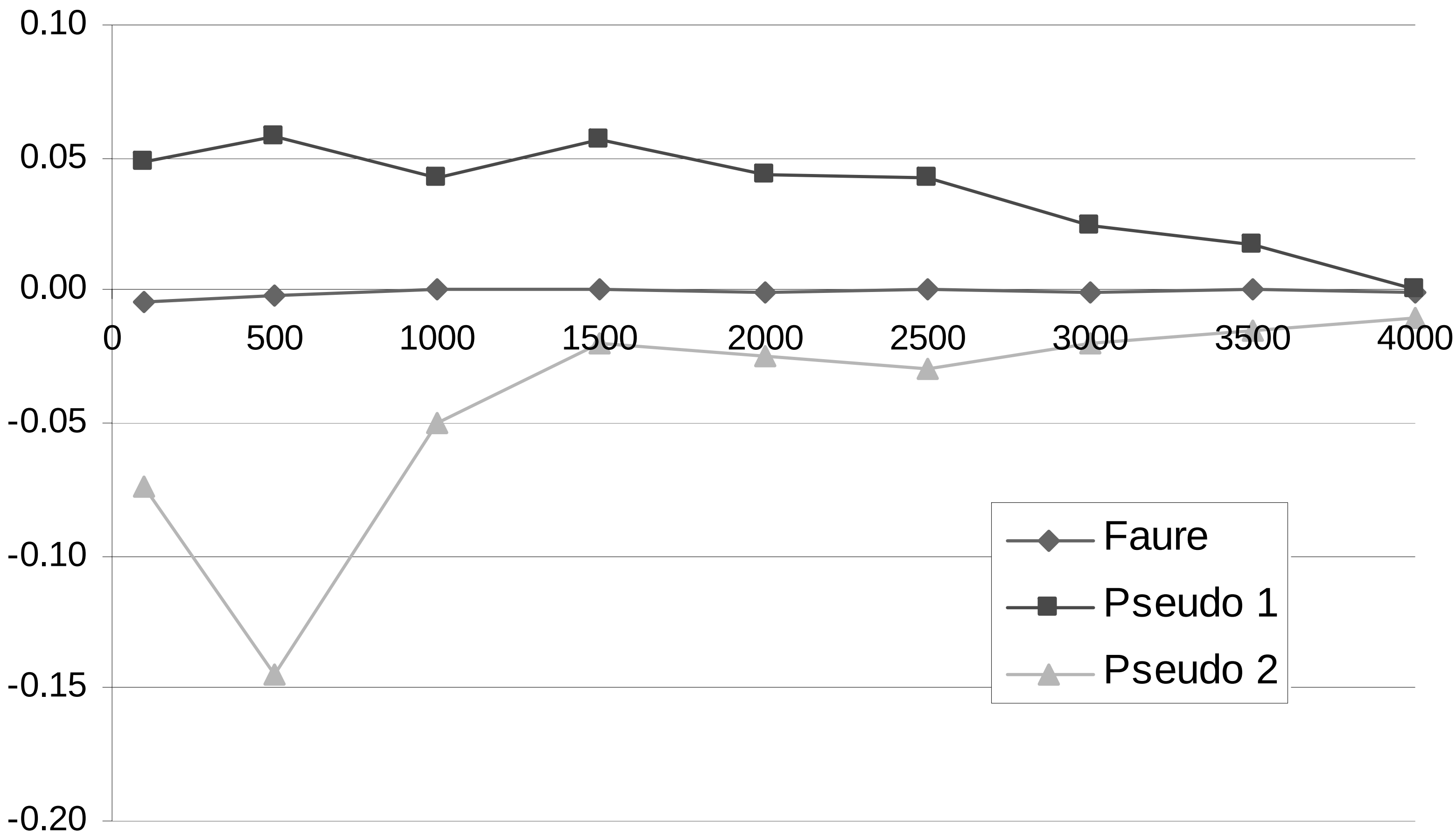


Random points
in the
unit square



Sobol points
in the
unit square

Pricing Error for a European Call



Sensitivity Factors with MCS

- Approximate using finite difference ratios
- Delta $$\frac{\partial C}{\partial P} \approx \frac{C(P+\Delta P) - C(P-\Delta P)}{2\Delta P}$$
- Gamma $$\frac{\partial^2 C}{\partial P^2} \approx \frac{C(P+\Delta P) - 2C(P) + C(P-\Delta P)}{\Delta P^2}$$
- Vega $$\frac{\partial C}{\partial \sigma} \approx \frac{C(\sigma+\Delta \sigma) - C(\sigma-\Delta \sigma)}{2\Delta \sigma}$$
- Theta $$\frac{\partial C}{\partial t} \approx \frac{C(t+\Delta t) - C(t-\Delta t)}{2\Delta t}$$

Summary

- Monte-Carlo Methods
- Random number generation
- Simulating diffusion process
- Applications to option pricing
- Variance reduction techniques