RMSC5102 Final Review

LEUNG MAN FUNG, HEMAN

SPRING, 2021

Agenda

Review

- Basic Knowledge
- The Black-Scholes World
- Monte Carlo Method
- Random Variable Generation
- Variance Reduction Technique
- Simulation in Action

Q&A

Basic Knowledge

Binomial Distribution

$X \sim \mathrm{B}(n, p)$

- Number of successes X in n independent Bernoulli trials
- Each trial has a probability of success p
- If $X_1, \dots X_n \sim \mathrm{B}(1, p)$, then $\sum_{i=1}^{n} X_i \sim \mathrm{B}(n, p)$ (sum of i.i.d. Bernoulli r.v.s is a binomial)

Pmf: $\mathbb{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$ for $x = 0, 1, 2, \dots, n$

Mgf: $M_X(t) = (1 - p + pe^t)^n$

Mean: $\mathbb{E}(X) = np$

Variance: $\mathrm{Var}(X) = np(1-p)$

Exponential Distribution

$X \sim \mathrm{Exp}(\lambda)$

- Continuous analogue of the geometric distribution
- We adopt the rate parametrization instead of scale

Pdf: $f(x) = \lambda e^{-\lambda x}$ for $x \geq 0$, otherwise 0

Cdf: $F(x) = 1 - e^{-\lambda x}$ for $x \geq 0$

Mean: $\mathbb{E}(X) = \frac{1}{\lambda}$

Variance: $\mathrm{Var}(X) = \frac{1}{\lambda^2}$

Memoryless property: $\mathbb{P}(X > s + t | X > s) = \mathbb{P}(X > t)$ for $s, t \geq 0$

- Exponential distribution is the only continuous distribution that has this property

Some Properties of Expectation, Variance and Covariance

Law of the unconscious statistician: $\mathbb{E}[g(X)] = \sum_{i=1}^{n} g(x_i)\mathbb{P}(X = x_i)$

- Continuous version: $\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x)f(x)dx$

Translation/rescale:

- $\mathbb{E}(aX + b) = a\mathbb{E}(X) + b$
- $\mathrm{Var}(aX + b) = a^2\mathrm{Var}(X)$
- $\mathrm{Cov}(aX + b, cY + d) = ac\mathrm{Cov}(X, Y), \mathrm{Cov}(X, X) = \mathrm{Var}(X)$
- $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\mathrm{Cov}(X, Y)$

Linearity of expectation: $\mathbb{E}(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathbb{E}(X_i)$

Alternative formula for variance: $\mathrm{Var}(X) = \mathbb{E}(X^2) - [\mathbb{E}(X)]^2$

Geometric Brownian Motion

SDE: $\mathrm{dS_t} = rS_t dt + \sigma S_t dW_t \Rightarrow \mathrm{S_t} = S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)t+\sigma W_t}$

- Use $S_T = S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)T+\sigma\sqrt{T}Z}$ in simulation to avoid simulating intermediate prices
- However, we need to store the intermediates price if the option is path dependent

Algorithm:

- 1) Generate $Z \sim \mathrm{N}(0,1)$
- 2) Set $S_T = S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)T+\sigma\sqrt{T}Z}$

Question may specify stock price dynamic other than GBM

- Use the given dynamic to simulate S_T like generating random variables

The Black-Scholes World

No-arbitrage Pricing

If two portfolios have the same payoff at time T, their costs should agree at time 0

- Otherwise, we can earn riskless profit by “buy low sell high”
- This assume no limit to arbitrage, e.g., we can short sell without margin

Basic argument in derivative pricing, e.g.,

- Put-call parity
- Black-Scholes formula

Black–Scholes–Merton Model

Black-Scholes formula

- $C(S_t, t) = \Phi(d_1)S_t - \Phi(d_2)Ke^{-r(T-t)}$,
- $P(S_t, t) = Ke^{-r(T-t)} - \mathrm{S_t} + C(S_t, t) = \Phi(-d_2)Ke^{-r(T-t)} - \Phi(-d_1)S_t$
- $d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right]$
- $d_2 = d_1 - \sigma\sqrt{T-t}$

Note that $P(S_t, t)$ is derived from put-call parity

- Put-call parity: $C_E - P_E = S - Ke^{-r(T-t)}$

Implied volatility

- Value of volatility when back-solving an option pricing model (such as BS) with current market price

Monte Carlo Method

Standard Monte Carlo

Vanilla European call option pricing

- Recall payoff function is $\max(S_T - K, 0)$
- Estimate $E[\max(S_T - K, 0)]$ by sample average $\frac{1}{n}\sum_{i=1}^{n} \max(S_T^{(i)} - K, 0)$

Algorithm

- 1) Generate $Z \sim \mathrm{N}(0,1)$
- 2) Set $S_T = S_0 e^{\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}Z}$
- **3) Compute $\boldsymbol{\pi_i = \max(S_T - K, 0)}$**
- 4) Repeat 1 to 3 for $i = 1, \ldots, n$
- 5) Option price = $\frac{\mathrm{e}^{-\mathrm{rT}}}{\mathrm{n}}\sum_{i=1}^{n} \pi_i$

Things to Note

General algorithm (always refer to tutorial notes)

- 1) Generate random variable X_i
- 2) Calculate $h_i = h(X_i)$, where h is the target function
- 3) Repeat 1 and 2 for n times
- 4) $\hat{\theta} = \frac{1}{n}\sum_{j=1}^{n} h_j$ (remember to do discounting if necessary)

Be careful of…

- X_i is not necessarily normally distributed
- The target function $h(x)$ that you are interested in
  - We need to adjust for conditional probability in stratified sampling sometimes because h changes
- How to generate X_i
  - For any X_i that does not follow $\mathrm{N}(0,1)$ or $\mathrm{U}(0,1)$. This includes discrete uniform

Random Variable Generation

Inverse Transform

Probability integral transform: if X is a continuous random variable with cdf F_X, then $Y = F_X(X) \sim \text{Unif}(0,1)$

- In other words, if we can find the cdf inverse F_X^{-1}, then $F_X^{-1}(U)$ and X have the same distributions

Algorithm (discrete)

- Generate $U \sim \text{Unif}(0,1)$
- $X = x_j$ if $\sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i$

Algorithm (continuous)

- Generate $U \sim \text{Unif}(0,1)$
- $X = F_X^{-1}(U)$ assuming the inverse exists

Rejection Sampling

If we can simulate $Y \sim G_Y$ easily, we can use the proportional distribution as a basis to simulate X with pdf $f(x)$

Algorithm

- 1) Find $c = \max_y \frac{f(y)}{g(y)}$
- 2) Generate Y_i from a density g, e.g., $U_1 \sim \text{Unif}(0,1) \Rightarrow Y_i = G^{-1}(U_1)$
- 3) Generate $U_2 \sim \text{Unif}(0,1)$
- 4) If $\text{U}_2 \leq \frac{1}{c} \cdot \frac{f(Y_i)}{g(Y_i)}$, set $X_i = Y_i$, otherwise return to 2

Inverse transform is rejection sampling with $c = 1$

- Because inverse transform simulate from F directly (always accept)

Variance Reduction Technique

Antithetic Variables

If we are able to generate negatively correlated underlying random variables, the estimator can have lower variance as compared with independent samples

- This requires the target function $h(x)$ to be monotone
- Show $h'(x) \geq 0$ or $h'(x) \leq 0$ within the target range for monotonicity
- As $h(x)$ is monotone, $\mathrm{Cov}[h(U), h(1-U)] \leq 0$ where $U \sim \mathrm{Unif}(0,1)$
- As half of your variables are antithetic, you only need to generate $\frac{n}{2}$ numbers for n samples

Useful corollary: if $h(x)$ is monotone, then $\mathrm{Cov}\big(h(U), h(1-U)\big) \leq 0$ where $U \sim \mathrm{Unif}(0,1)$

Antithetic Variables

Algorithm:

- 1) Generate $U \sim \text{Unif}(0,1)$
- 2) Set $X_i = F^{-1}(U), Y_i = F^{-1}(1-U)$ (note: want X, Y same distribution but negative correlation)
- 3) Repeat 1 and 2 for $i = 1, \ldots, n$
- 4) $\hat{\theta} = \frac{1}{2n}\sum_{i=1}^{n}[h(X_i) + h(Y_i)]$

Note:

- $F^{-1}(U)$ is monotone in general as cdf is monotone
- Hence $h[F^{-1}(U)]$ is monotone if $h(\cdot)$ is monotone

Stratified Sampling

If we have information about grouping in the population, then we may use conditional mean (mean of subgroup) as the sample from the population

Algorithm:

- 1) Generate $V_{i,j} = \frac{1}{B}(U_{i,j} + i - 1)$ where $U_{i,j} \sim \text{Unif}(0,1)$ for $i = 1, \dots, B; j = 1, \dots, N_B$
- 2) Set $X_{i,j} = F^{-1}(V_{i,j})$
- 3) $\hat{\theta} = \frac{1}{B \times N_B} \sum_{j=1}^{N_B} [h(X_{1,j}) + \text{h}(X_{2,j}) + \cdots + \text{h}(X_{B,j})]$ (average over subsamples and bins, remember to adjust for conditional probability)

Note:

- i represents index of bins and j represent index of elements within a bin

Stratified Sampling

When to adjust for conditional probability?

- Try to write out the expectation you are trying to approximate (e.g. Ch6 p.43)
- Usually you need to when there is an indicator or you have restricted the support

Control Variate

If we combine the estimate of our target unknown quantity with estimates of some known quantities, we can exploit the known information

Algorithm:

- 1) Find μ_Y for Y with a known distribution (or estimate μ_Y via pilot simulation)
- 2) Generate X_i, Y_i for $i = 1, \dots, n$
- 3) Compute $\bar{X}, \bar{Y}, \hat{\sigma}_{X,Y}, \hat{\sigma}_Y^2$
- 4) $\hat{\theta} = \bar{X} - \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_Y^2}(\bar{Y} - \mu_Y)$

Pilot simulation: we can run a simulation with small sample size (e.g. $m = 100$) and compute $\hat{\sigma}_{XY}, \hat{\sigma}_Y^2$ and $\mu_Y = \bar{Y}_m$ based on this pilot sample. Then we can use their values when we compute $\hat{\theta} = \bar{X}_n - \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_Y^2}(\bar{Y}_n - \mu_Y)$ for our target n samples

Control Variate

Properties of effective control: evaluable from simulation data, known mean and high correlation with the simulation variable. Possible candidates are underlying random variable (e.g. uniform when we use inverse transform) and martingale transform (will not be tested)

Note:

- The algorithm in last slide is one-off, i.e. it does not affect each sample
- So we should use control variate last if we were to combine the methods

Importance Sampling

If certain values of the simulation variable have more impact on the parameter of interest (e.g. probability of a rare event)

- We can try to “emphasize” those values by sampling them more frequently and reduce variance
- This can be done by changing the probability measure using the likelihood ratio as weight

Algorithm:

- 1) Find the likelihood ratio $\frac{f(x)}{g(x)}$ where $f(x)$ is the original target pdf
- 2) Generate $X_i \sim G$ for $i = 1, \dots, n$
- 3) $\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} \frac{h(X_i)f(X_i)}{g(X_i)}$

Maximum principle: choose g such that both $g(x)$ and $h(x)f(x)$ take maximum values at the same $x = x^*$ (not in our syllabus, just for your reference)

Importance Sampling

A possible candidate for $g(x)$ is the tilted density

Tilted density: $f_t(x) = \frac{e^{tx} f(x)}{M_X(t)}$ where $M_X(t)$ is the moment generating function of X

Choice of t in importance sampling

- Find the upper bound $\frac{h(x)f(x)}{f_t(x)} \leq \frac{h(x_t^*)f(x_t^*)}{f_t(x_t^*)}$ and minimize for t
  - The upper bound should only depends on t for minimization
- In other words, find $x = x_t^*$ such that $\frac{h(x)f(x)}{f_t(x)} \leq \frac{h(x_t^*)f(x_t^*)}{f_t(x_t^*)}$ for all x in the support
  - x_t^* has subscript t because it may depend on t
- Then minimize $\frac{h(x_t^*)f(x_t^*)}{f_t(x_t^*)}$ with respect to t

Simulation in Action

Down-and-in Call Option

Price of down-and-in option: $C_{di} = e^{-rT}\mathbb{E}\left[(S_T - K)^+ \mathbb{I}\left(\min_{0\le t\le T} S_t < V\right)\right]$

Algorithm:

- 1) Generate $Z \sim \mathrm{N}(0,1)$
- 2) Set $S_{t_i} = S_{t_{i-1}} \exp\left[\left(r - \frac{1}{2}\sigma^2\right)(t_i - t_{i-1}) + \sigma\sqrt{t_i - t_{i-1}}Z\right]$
- 3) Repeat step 1 and 2 for $i = 1, \dots, n$ where $t_0 = 0$ and $t_n = T$
- 4) If $\min_i S_{t_i} < V$, set $C_j = e^{-rT}\max(S_T - K, 0)$. Otherwise set $C_j = 0$
- 5) Repeat step 3 and 4 for $j = 1, \dots, N$
- 6) The price is given by $\hat{\theta} = \frac{1}{N}\sum_{j=1}^{N} C_j$

Down-and-in Call Option

What if you already have the price of a vanilla call of the same parameters?

- i.e. $C_v = e^{-rT}\mathbb{E}[(S_T - K)^+]$
- Possible to evaluate $\mathbb{E}\left[\mathbb{I}\left(\min_{0\le t\le T} S_t < V\right)\right]$ and combine with C_v directly
  - This probably will not be tested. Just to stimulate your thinking
- Your target function h becomes $\mathbb{I}\left(\min_{0\le t\le T} S_t < V\right)$ in that case
- This is kind of like adjusting for conditional probability

Path-dependent Option

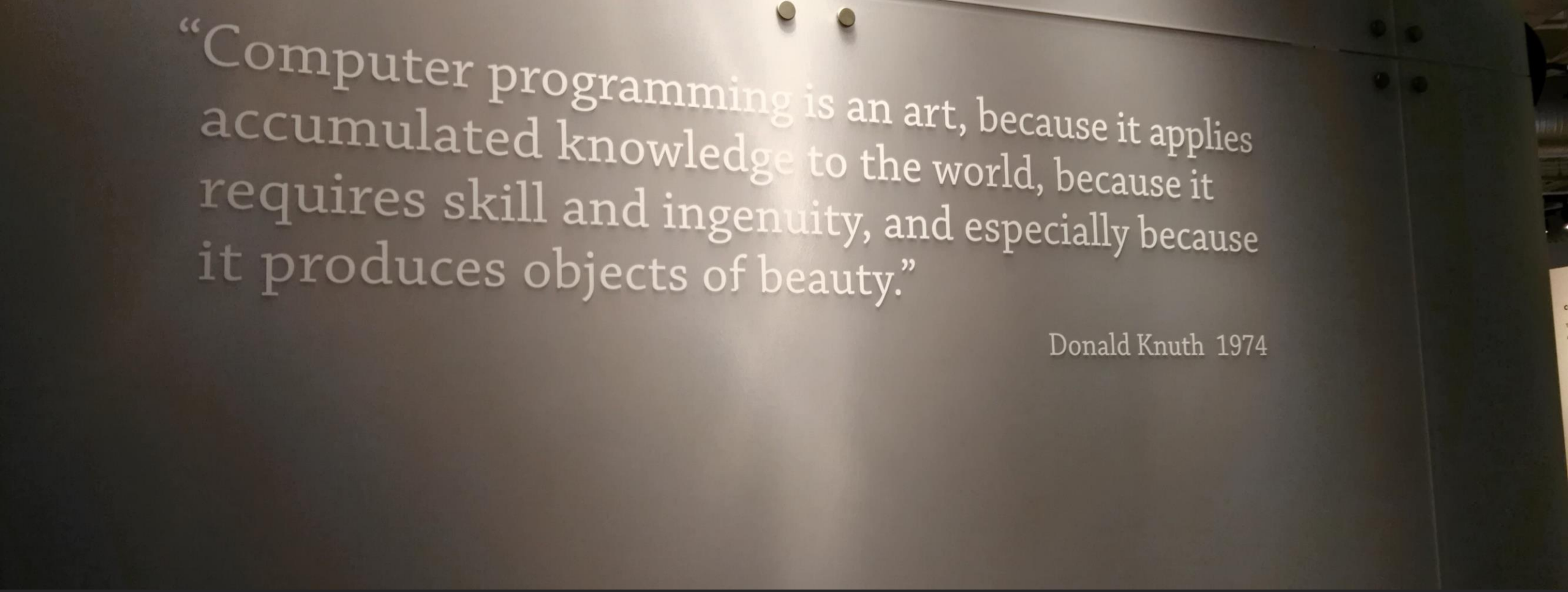
Problem with discretization

- The discretized process does not have the correct transition density
  - First order Euler scheme has normal increments
  - Second order Milstein scheme has non-central chi square increments
  - Optimal tradeoff between n and N exists for the two schemes. See Duffie and Glynn (1995)
    - Content of the paper will not be tested
- The discretized process may incorrectly evaluate the payoff
  - E.g., Asian option
  - Possible solution: Brownian bridge. See Beaglehole, Dybvig and Zhou (1997)

Path-dependent Option

American option

- Problem with branching paths
- Possible solution: linear regression. See Longstaff and Schwartz (2001)
- Just for your interest
  - Consider taking courses from Prof. Wong :P He taught us this in undergrad



Q&A

Thank you for taking RMSC5102! Write me (or department) an email if you like my tutorials :P

Let’s keep in touch :)