Financial Engineering

Zehra Eksi

Institute for Statistics and Mathematics
Dept. of Finance, Accounting and Statistics

Spring 2019

© 2014 BOANET.AT

Part I

Introduction

Outline

- Organization
- Financial Engineering
- Literature

Course Info

Title:	Financial Engineering (5195)
Instructor:	*Assoc. Prof. Zehra Eksi*
Contact details:	zehra.eksi@wu.ac.at
Office hours:	Tue $13:30-15:00$

- Lectures: Tuesdays from 9:00 to 13:00
- Attendance: mandatory (attend at least %80 of all lectures, i.e., at most one out of seven sessions can be missed)

Assessment and Date of Exams

- Weekly homework assignments (%30):
  - Submission as a group of at most four people (%20)
  - Presentation of solutions in the class by one of the group members (%10)
- Two written exams (%60):
  - mid-term (%20): 26.03.2019, TC.4.01
  - final (%40): 30.04.2019, 9:00-11:00, TC.5.15

Prerequisites and Objective

- Prerequisites:
  - Knowledge in (continuous-time) finance
  - Some knowledge in statistics, probability and stochastic processes
  - Knowledge of a programming language
- Main goal: to become familiar with the essential techniques and tools for *financial engineering*
- Material: lecture slides will be updated continously (available at Learn@WU)

Outline

- Organization
- Financial Engineering
- Literature

What is financial engineering?

Financial engineering is...

an interdisciplinary area consisting of finance, engineering, tools of mathematics and the practice of programming.

The main applications of financial engineering are to:

- portfolio management
- risk management
- financial regulation ⇑
- structured products ⇓
- derivatives pricing ⇓
- trading and execution ⇑

For ⇓ visit *http://blogs.reuters.com/emanuelderman/2011/07/07/financial-engineering-as-a-career-part-1/*

Topics to be covered

- Principles of derivatives pricing;
- Principles of Monte Carlo;
- Generating random variables and stochastic processes;
- Simple variance reduction techniques;
- Pricing exotic (Bermudan) options by means of Monte Carlo simulation;
- Construction of yield-curve;
- Applications in risk management.

Outline

- Organization
- Financial Engineering
- Literature

References

- Paul Glasserman [PG] : Monte Carlo Methods in Financial Engineering (2004)
- Rüdiger Seydel [RS]: Tools for Computational Finance (2009)
- Paolo Brandimarte [PM]: Handbook in Monte Carlo Simulation: Applications in Financial Engineering, Risk Management, and Economics (2014)
- Damiano Brigo and Fabio Mercurio [BM]: Interest rate models-theory and practice: with smile, inflation and credit (2007)
- Damir Filipovic [DF]: Term Structure Models (2009)

Part II

Principles of Derivatives Pricing

EFMD EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION AMBA ACCREDITED

Outline

- Main ideas
- Approaches to Derivatives Pricing

Derivative Instruments

Definition

A derivative is an instrument whose value is derived from the value of one or more underlying assets.

Some examples:

options (European, American, Bermudan option...); futures; forwards; swaps...

Undelying assets include

stocks; bonds; commodities; currencies; weather; inflation; credit risk...

Pricing of Derivatives

- Pricing derivatives constitute an important place in financial engineering.
- Given the structure of the contract and the price of the underlying, the objective is to find the *fair price*.
- Mostly, the idea of "no arbitrage" yields the *fair price*: The price of a derivative security should be equal to the cost of perfectly replicating the security through trading in other assets.

Three main principles to keep in mind

P1 If a derivative security can be perfectly replicated through trading in other assets (existence of a self-financing replicating strategy), then the price of the derivative is the cost of replication.

P2 Discounted asset prices are martingales under a probability measure associated with the choice of discount factor (or numeraire).

P3 In a *complete* market, any payoff can be replicated through a trading strategy, and the martingale measure associated with a numeraire is unique.

Outline

- Main ideas
- Approaches to Derivatives Pricing

Approaches to derivative pricing

PDE Approach

[P1] together with the given dynamics of the underlying asset lead to a partial differential equation (PDE) that the price of the derivative satisfies.

Risk-Neutral (Martingale) Approach

[P2] gives us a way to express the price of the derivative as the expected present value of the terminal payoff discounted at the risk-free rate.

Naturally, results of the two approaches should coincide !

A more technical link through the two approaches I

Feynman-Kac formula

Consider function $\mu(x)$, $\sigma(x)$, $r(x)$ and some function ϕ on $\mathbb{R}$. Suppose that V(t,x) solves the terminal value problem

$$\begin{aligned} &\frac{\partial V}{\partial t}(t,x) + \mu(x)\frac{\partial V}{\partial x}(t,x) + \frac{1}{2}\sigma(x)^2\frac{\partial^2 V}{\partial x^2}(t,x) = r(x)V(t,x), \\ &V(T,x) = \phi(x). \end{aligned} \tag{1}$$

Then, it holds for $t_0 \leq T$ that

$$V(t_0,x) = \mathbb{E}_x\left(\exp\left(-\int_0^{T-t_0} r(X_s)ds\right)\phi(X_{T-t_0})\right), \tag{2}$$

where X solves the SDE

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t,\, X_{t_0} = x. \tag{3}$$

A more technical link through the two approaches II

Feynman-Kac formula can be used in two ways:

- Compute the expectation (2) in order to solve numerically the PDE in (1).
- We can solve (numerically) the PDE in (1) to compute the expectation in (2) .

Possible problems

- PDE Approach:
  - A solution may not exist when underlying asset price dynamics are complex.
  - Numerical solution may be impractical when number of underlyings for replication is large.
- Risk-neutral Approach:
  - Most of the time it is not possible to calculate the expectation (integral) explicitly.
  - Standard numerical solution techniques may be impractical when number of underlyings is large.

Possible solution...

We can use *Monte Carlo* simulation to compute the expectation numerically.

Part III

Principles of Monte Carlo Simulation

EFMD EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION AMBA ACCREDITED

Outline

- Monte Carlo Integration
- Generating Random Variables
- Simulating Poisson Process
- Simulating Brownian Motion

Monte Carlo Integration I

- Suppose we want to compute

$$\Theta = \int_0^1 g(x)dx$$

- It may not be possible to compute analytically.
- We make the observation that

$$\Theta = \mathbb{E}(g(U))$$

where $U \sim U(0,1)$.

- Given a $U(0,1)$ random number generator, this gives a way to estimate Θ via:
  1. Generate IID sample $U_1, U_2, \ldots, U_n$ from $U(0,1)$,
  2. Compute

$$\hat{\Theta}_n = \frac{g(U_1) + g(U_2) + \cdots + g(U_n)}{n}.$$

- Is $\hat{\Theta}_n$ a good estimator of Θ?

Monte Carlo Integration II

Properties of $\hat{\Theta}_n$

$\hat{\Theta}_n$ is an *unbiased* and *consistent* estimator of Θ, i.e.,

- $\mathbb{E}(\hat{\Theta}_n) = \Theta$,
- $\hat{\Theta}_n \to \Theta$ as $n \to \infty$, a.s. This is a direct consequence of the Strong Law of Large Numbers (SLLN).

Recall: SLLN

Let $X_1, X_2, \ldots$ be iid random variables with $\mathbb{E}(X_i) = \mu$ and $var(X_i) = \sigma^2 < \infty$, and define $\bar{X}_n = (1/n)\sum_{i=1}^{n} X_i$. Then, for every $\epsilon > 0$,

$$P(\lim_{n\to\infty} |\bar{X}_n - \mu| < \epsilon) = 1;$$

that is $\bar{X}_n \xrightarrow{a.s.} \mu$

Monte Carlo Integration III

Example

- Compute the integral $\Theta = \int_2^4 (x^3 + x)dx$ by Monte Carlo method with $n = 10000$.
  - Notice that $\Theta = 2\int_2^4 (x^3 + x)\frac{1}{2}dx$.
  - That is, for $X \sim U(2, 4)$, we have $\Theta = 2\mathbb{E}(X^3 + X)$
  - Hence we can estimate Θ by generating 10000 IID $U(0, 1)$, transforming this into (HOW?) IID $U(2, 4)$ random variables $X_1, X_2, \ldots, X_{10000}$ and then computing $\hat{\Theta}_n = \frac{2}{n}\sum_{i=1}^{n}(X_i^3 + X_i)$
- One can actually compute the integral analytically. We have $\Theta = 66$.
- How close are the two results?

Monte Carlo Integration IV

Algorithm: Monte Carlo Integration

Given inputs g, interval (a, b), sample size n

1. for $i = 1 : n$
2. generate $U_i \sim U(0, 1)$
3. transform $X_i \leftarrow (b - a)U_i + a$
4. $Y_i \leftarrow g(X_i)$
5. end for
6. $\hat{\Theta} \leftarrow \frac{(b-a)}{n} \sum_{i=1}^{n} Y_i$

Monte Carlo Error I

Monte Carlo error

The Monte Carlo error (MCE) for a given number of simulation trials n is defined as the difference between the estimate Θ_n and Θ:

$$MCE := \Theta_n - \Theta$$

- The error depends on the sample, hence it is random as well.
- But we can characterize the distribution of the MCE by the help of CLT.

Monte Carlo Error II

Theorem (Central Limit Theorem)

Let $X_1, X_2, \ldots$ be a sequence of iid random variables with $\mathbb{E}(X_i) = \mu$ and $0 < var(X_i) = \sigma^2 < \infty$. Define $\bar{X}_n = (1/n)\sum_{i=1}^{n} X_i$. Let $G_n(x)$ denote the cdf of $\sqrt{n}(\bar{X}_n - \mu)/\sigma$. Then, for any x, $-\infty < x < \infty$,

$$\lim_{n\to\infty} G_n(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy;$$

That is, $\frac{(\bar{X}_n - \mu)}{\sigma/\sqrt{n}}$ (standardized sample means) has a limiting standard normal distribution.

Monte Carlo Error III

- From the central limit theorem it follows that the MCE converges in distribution to $N(0, \frac{\sigma}{\sqrt{n}})$.
- The term $\sigma/\sqrt{n}$ referred to as the standard error.
- Notice that cutting the error in half requires to quadruple the number of simulations (n) .
- Adding one decimal place of precision requires 100 times as many simulations.

Multi-Dimensional Monte Carlo Integration

- We consider the problem

$$\Theta = \int_0^1 \int_0^1 g(x,y)dxdy.$$

- Recall that for $U^{(1)}$, $U^{(2)}$ independent $U(0,1)$ random variables we have $f(u^1,u^2) = f_1(u^1)f_2(u^2) = 1$ on $(0,1)^2$.
- Hence, we can write $\Theta = \mathbb{E}(g(U^{(1)}, U^{(2)}))$.
- To estimate Θ:
  - generate n of U(0,1) random vectors $(U_i^{(1)}, U_i^{(2)})$
  - $\hat{\Theta}_n = \frac{g(U_1^{(1)},U_1^{(2)})+g(U_2^{(1)},U_2^{(2)})+\cdots+g(U_n^{(1)},U_n^{(2)})}{n}$
- $\hat{\Theta}_n$ still preserves the desirable properties.

Monte Carlo integration for more general problems

- Suppose now we want to compute

$$\Theta = \int \int_D g(x,y) f(x,y) dx dy$$

  where $f(x,y)$ is some density on D.
- Hence we have $\Theta = \mathbb{E}(g(X,Y))$ where (X,Y) has the joint density $f(x,y)$.
- To estimate Θ we can generate n random vectors (X,Y) from the joint density $f(x,y)$ and compute

$$\hat{\Theta}_n = \frac{g(X_1,Y_1) + g(X_2,Y_2) + \cdots + g(X_n,Y_n)}{n}$$

Outline

- Monte Carlo Integration
- Generating Random Variables
- Simulating Poisson Process
- Simulating Brownian Motion

Methods for Generating Random Variables I

There are three main methods to generate random variables:

- the inverse transform method
- the composition method
- the acceptance-rejection method

For the Inverse transform method, we mainly make use of the following well-known result:

Theorem (Probability integral transformation)

Let X have cdf $F(x)$ and define the RV Y as $Y = F(X)$. Then Y is uniformly distributed on $(0, 1)$, that is, $P(Y \leq y) = y$, $0 < y < 1$.

Methods for Generating Random Variables II

Inverse Transform Method

- We want to sample from a CDF F, i.e., to generate a random variable X with $\mathbb{P}(X \leq x) = F(x)$
- This method sets $X = F^{-1}(U),\ U \sim U(0,1)$.
- Hence

$$\begin{aligned}\mathbb{P}(X \leq x) &= \mathbb{P}(F^{-1}(U) \leq x) \\ &= \mathbb{P}(U \leq F(x)) \\ &= F(x).\end{aligned}$$

- If the inverse of F is not well-defined we may set

$$F^{-1}(u) = \inf\{x : F(x) \geq u\}.$$

Methods for Generating Random Variables III

Example: exponential distribution-inverse transform method

- We wish to generate $X \sim exp(\lambda)$.
- We have the cdf $F(x) = 1 - e^{-\lambda x}$, $x \geq 0$
- Hence, $F^{-1}(u) = -\log(1-u)/\lambda$.
- To sample from $exp(\lambda)$:
  i. Generate $U \sim U(0,1)$;
  ii. Set $X = -\log(u)/\lambda$ (WHY?).

Methods for Generating Random Variables IV

Example: discrete distributions-inverse transform method

- Suppose we have a discrete random variable with possible values $c_1 < \cdots < c_n$.
- Let p_i be the probability associated to c_i
- Set $q_0 = 0$, and $q_i = \sum_{j=1}^{i} p_j$, $i = 1, 2, \ldots, n$ (Hence $q_i = F(c_i)$).
- To sample from this distribution
  i. generate $U \sim U(0, 1)$
  ii. find $K \in \{1, \ldots, n\}$ s.t. $q_{K-1} < U \leq q_K$
  iii. set $X = c_K$.

Methods for Generating Random Variables V

The Composition Method

- Suppose we have $X \sim F$ and we can write $F(x) = \sum_{i=1}^{\infty} w_i F_i(x)$, where $w_i \geq 0$ and $\sum_i w_i = 1$ and F_is are cdfs.
- We may often have such representations, e.g., $Hyperexp(\lambda_1, \alpha_1 \ldots, \lambda_n, \alpha_n)$ with

$$f(x) = \sum_{i=1}^{n} \alpha_i \lambda_i e^{-\lambda_i x}$$

- *How can we show that this method actually works?*
- We can make use of the following algorithm:
  i. Generate K that is distributed on the positive integers s.t $\mathbb{P}(K = j) = w_j$. (How can we do this?)
  ii. If $K = j$, then generate Z_j from the cdf F_j;
  iii. Set $X = Z_j$.

Methods for Generating Random Variables VI

Acceptance-Rejection Method

- Suppose we want to generate sample for a rv X with density f and cdf, F.
- Suppose it's hard to simulate a value of X directly using inverse transform or composition algorithms.
- Let Y be another rv with density g and suppose it's easy to simulate Y.
- If there exists a constant c such that $f(x) \leq cg(x)$, for all x, then we can simulate a value of X as:

  i generate Y from distribution g
  ii generate $U \sim U(0,1)$
  iii if $U \leq f(Y)/cg(Y)$
  return X
  otherwise
  go to Step(i).

Methods for Generating Random Variables VII

Generating Multivariate Normals

- Suppose we want to generate the random vector $X = (X_1, \ldots, X_n)$ where $X \sim N_n(0, \Sigma)$.
- Let $Y = (Y_1, \ldots, Y_n)$ where Y_is are IID $N(0, 1)$.
- If A is an $n \times n$ matrix then

$$Z = AY \sim N_n(0, AA^\top)$$

- We can generate independent Normal rvs $Y_1, \ldots, Y_n$ and consider them as a vector.
- Thus, the problem of sampling from X reduces to finding a matrix A s.t. $AA^\top = \Sigma$

Methods for Generating Random Variables VIII

Cholesky factorization

- Among all possible $\boldsymbol{A}$, a lower triangular one is obtained as a result of Cholesky factorization
- However, be careful if Σ is positive semi-definite (hence rank deficient).
- In this case it is better to reduce the problem to one of full rank, find subvector $\tilde{\boldsymbol{X}}$ and matrix $\boldsymbol{D}$ s.t the covariance matrix $\tilde{\Sigma}$ is full rank and that

$$\boldsymbol{D}\tilde{\boldsymbol{X}} = \Sigma.$$

- Cholesky factorization can now be applied to $\tilde{\Sigma} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^{\top} \Rightarrow \boldsymbol{X} = \boldsymbol{D}\tilde{\boldsymbol{A}}\boldsymbol{Y}$.
- Such a situation may occur. e.g., in case of factor models in which the vector $\boldsymbol{X}$ of length $\boldsymbol{n}$ is determined by $\boldsymbol{k} < \boldsymbol{d}$ number of risk sources (factors).

Outline

- Monte Carlo Integration
- Generating Random Variables
- Simulating Poisson Process
- Simulating Brownian Motion

Simulating Poisson Process I

Poisson Process

Let $(\tau_i)_{i\geq 1}$ be a sequence of independent exponential random variables with parameter λ and $T_n = \sum_{i=1}^{n} \tau_i$. The process $\{N_t, t \geq 0\}$ defined by

$$N_t = \sum_{n\geq 1} I_{\{t\geq T_n\}}$$

is called a Poisson process with intensity λ.

Simulating Poisson Process II

- For a Poisson process the numbers of arrivals in non-overlapping intervals are independent and the distribution of the number of arrivals in an interval only depends on the length of the interval.
- It is a counting process with

$$\mathbb{P}(N_t = r) = \frac{(\lambda t)^r e^{-\lambda t}}{r!}.$$

- From its definition, one can simulate a Poisson process by simply generating the $exp(\lambda)$ inter-arrival times, τ_i.

Simulating Poisson Process III

Simulation Algorithm: Poisson Process

set $t = 0$, $I = 0$

generate $U \sim U(0, 1)$

set $t = t - \log(U)/\lambda$

while $t < T$

    set $I = I + 1$, $S(I) = t$

    generate $U \sim U(0, 1)$

    set $t = t - \log(U)/\lambda$

Outline

- Monte Carlo Integration
- Generating Random Variables
- Simulating Poisson Process
- Simulating Brownian Motion

One-dimensional standard Brownian motion

Brownian motion

One-dimensional standard Brownian motion on $[0, T]$ is a stochastic process $\{W(t), 0 \leq t \leq T\}$ with the following properties:

i. $W(0) = 0$;

ii. the mapping $t \mapsto W(t)$ is, with probability 1, a continuous function on $[0, T]$;

iii. the increments $\{W(t_1) - W(t_0), W(t_2) - W(t_1), \cdots, W(t_k) - W(t_{k-1})\}$ are independent for any k and any $0 \leq t_0 < t_1 < \cdots < t_k \leq T\}$,

iv. $W(t) - W(s)$ is distributed as $N(0, t - s)$ for any $0 \leq s < t \leq T$.

Note that from i. and iv. $W(t) \sim N(0, t)$. Also for constants μ and $\sigma > 0$, we call process (X_t) a Brownian motion with drift μ and diffusion coefficient σ^2 if

$$\frac{X(t) - \mu t}{\sigma}$$

is a standard BM by setting $X(t) = \mu t + \sigma W(t)$.

Random walk construction

- In discussing the simulation of BM, we can focus on simulating values $(W_{t_0}), \ldots, W_{t_n})$ at a fixed set of points $0 < t_1 < \ldots < t_n$, since BM has independent normally distributed increments.
- Let $Z_1, \ldots, Z_n$ be independent standard normal variables, then for a standard BM, with $t_0 = 0$ and $W(0) = 0$, we can generate

$$W(t_{i+1}) = W(t_i) + \sqrt{t_{i+1} - t_i} Z_{i+1}, \quad i = 0, \ldots, n-1.$$

- The vector $(W_{t_1}), \ldots, W_{t_n})$ is a linear transformation of the vector of increments $\{W(t_1) - W(t_0), W(t_2) - W(t_1), \cdots, W(t_n) - W(t_{n-1})\}$, and since these increments are independent and normally distributed we can conclude that $(W_{t_1}), \ldots, W_{t_n})$ has a multivariate normal distribution.

Simulation with Cholesky Factorization

- Note that for simulating the multivariate normal, we need mean vector and the covariance matrix.
- From the independent increments property one can show that for $s \leq t$ $Cov(W(s), W(t)) = s$, and let C denote the covariance matrix of $(W_{t_1}), \ldots, W_{t_n})$, with the entries $C_{ij} = \min(t_i, t_j)$.
- $(W_{t_1}), \ldots, W_{t_n})$ has the distribution $N(0, C)$ and one can simulate this vector as AZ, where $Z = (Z_1, \ldots, Z_n)^\top \sim N(0, I)$ and A satisfies $AA^\top = C$
- The Cholesky factorization for C yields the lower triangular matrix A given by

$$A = \begin{bmatrix} \sqrt{t_1} & 0 & \cdots & 0 \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \cdots & \sqrt{t_n - t_n - 1} \end{bmatrix},$$

Simulation of Geometric Brownian Motion

Definition:Geometric Brownian Motion

A stochastic process $\{X_t : t \geq 0\}$, is a geometric Brownian motion (GBM) with drift μ and volatility σ if

$$\log(X) \sim BM(\mu - \frac{\sigma^2}{2}, \sigma).$$

That is

$$X_t \sim \log N((\mu - \frac{\sigma^2}{2})t, \sigma^2 t)$$

- Question: How would you simulate X_{t_i}?