Chapter 3

Low Discrepancy Sequences and Simulation of Stochastic Processes (v.A.3)

Low Discrepancy Sequences and Quasi Monte Carlo Simulation

The Monte Carlo simulation method is based on the idea that the random variates we generate are truly (or close to) random. That is, the uniformly distributed random numbers we obtain by using generators are truly random, so that a sample of uniforms that we generate on [0.1] will be "evenly" spread on the [0,1] interval. However, the random numbers produced by any of the algorithms we considered are not random and when we use them to estimate integrals or expectations of random variables, our estimates contain errors.

When generating random numbers, the idea is to generate a sequence of numbers that are uniformly spread over the [0,1] interval in a one-dimensional case. If a large enough number of uniform variates are generated, then one may argue that the empirical distribution of those variates is reasonably close to the Uniform [0,1] distribution. However, it is computationally costly to generate a large number of variates in order to control the error. Therefore, it is important to be able to reduce the computational cost of simulations (the number of variates that are needed to achieve certain errors) and keep errors under control.

In this section, we will use an algorithm to sequentially generate a sequence of numbers that are "uniformly" distributed over the [0,1] interval, regardless of the number of variates we generate. That is, no matter how many variates we take, they are uniformly (or close to uniformly) distributed over the [0,1] interval. That is, we would like to use an algorithm, in which the next generated number is placed on the unit interval in such a way that the distribution of the sequence is as close to uniform as possible at any step.

First, we will define the discrepancy of a sequence of numbers. Assume we want to generate a sequence of N random vectors $(X_1, X_2, \dots, X_N)$ on the n-dimensional hypercube I^n.

$$I^n = [0,1] \times [0,1] \times \dots \times [0,1] \subset \mathbb{R}^n.$$

If the vectors are uniformly distributed on I^n, then, by intuition, the number of variates in any subset A of I^n should be roughly proportional to the volume of A, relative to the volume of I^n. Define the set of all rectangular subsets of I^n:

$$A = \{[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n], \text{for any } 0 \le a_i \le b_i \le 1, i = 1,2, \dots, n\}$$

Define the discrepancy $D(x, A)$ of a set of points $x = (x_1, x_2, \dots, x_m)$ with respect to set A as follows:

$$D(x, A) = \sup_{B \in A} \left| \frac{Number\ of\ x_i\ in\ B}{m} - Volume\ of\ B \right|$$

Here $x_i \in \mathbb{R}^n$. A sequence would be "close-to-uniform", if the discrepancy of the sequence is small. The lower the discrepancy, the closer the sequence is to uniformity. We will describe a few algorithms that allow users generate low discrepancy sequences.

How would one use these sequences in simulation analysis?

When computing integrals, one can use sequences of low-discrepancy numbers instead of the pseudo-random sequences, as in Monte Carlo simulations. Theoretical results suggest that low-discrepancy sequences may perform better than pseudorandom sequences when estimating integrals.

The use of low-discrepancy sequences would be justified if we could prove their faster convergence (or lower variance) compared to the standard Monte Carlo method. It is known that the convergence rate for the Monte Carlo method is in the order of $n^{-1/2}$. Is has been shown that the convergence rate when low-discrepancy sequences are used is much better, in the order

of a $\frac{\ln(n)^N}{n}$, where N is the dimension of the problem. As it is easy to see, this is a better rate of convergence than $n^{-(1-\varepsilon)}$ for any small $\varepsilon > 0$.

Definition: A sequence $x_1, x_2, \ldots, x_m, \ldots \in \mathbb{R}^n$ is said to be a Low Discrepancy Sequence (LDS), if there exists a finite constant K_m so that the Discrepancy of the Sequence satisfies

$$D_n \leq K_m \frac{(\ln n)^m}{n} \text{ for any } n.$$

We will use a few defined low discrepancy sequences to estimate integrals. Using Quasi Monte Carlo methods, we can approximate integrals as follows:

Suppose $h_1, h_2, \ldots, h_k, \ldots$ are from a Low Discrepancy Sequence. Then, we set

$$\int_0^1 f(x)\, dx \approx \frac{1}{k} \sum_{i=1}^{k} f(h_i)$$

We will implement one of the popular LDS - the Halton's low discrepancy sequence. We refer interested readers to the book by P. Glasserman, for more details about other sequences (such as Sobol, Feure, Van der Corput, etc.) and for proofs of convergence results.

Comment:

To compute multi-dimensional integrals using the LDS we follow the same technique:

$$\int_0^1 \int_0^1 f(x,y) dx\, dy \approx \frac{f(H_1^1, H_2^1) + f(H_1^1, H_2^1) + \cdots + f(H_1^n, H_2^n)}{n}$$

where $H_1^k \; and H_2^k$ are the k-th numbers of Halton sequences with different bases (explained below).

Halton's Low-Discrepancy Sequences

Define $H(k,m)$ to be the Halton's $k-th$ number in the sequence with base m. The algorithm for generation of $H(k,m)$ is as follows:

STEP 1: For any $k \geq 1$ integer, write $k = a_0 + a_1 m + \cdots + a_r m^r$, where $a_i \in \{0,1, \ldots, m-1\}$, and r is so that $m^r \leq k < m^{r+1}$.

STEP 2: Define $H(k,m) = \frac{a_0}{m} + \frac{a_1}{m^2} + \cdots + \frac{a_r}{m^{r+1}}$ -- the $k\ th$ member of the Halton's sequence with base m.

Comment: m is chosen as a prime number.

For estimation of multidimensional integrals, sometimes one might need to generate multidimensional Halton sequences. To generate n −dimensional Halton sequences, we follow the following steps:

STEP 1: Choose the first n prime numbers: $m_1, m_2, \ldots, m_n$. These would be the bases for the Halton's sequences.

STEP 2: For any $k \geq 1$ integer, and m_i, using the method above, generate the vector of Halton's numbers: $X_k = (H(k,m_1), H(k,m_2), \ldots, H(k,m_n))$

The following Matlab function generates a one-dimensional Halton sequence (the code is adopted from Brandimarte's "Numerical methods in finance and economics: a MATLAB-based introduction")":

```
function Seq = GetHalton(HowMany, Base)
Seq = zeros(HowMany,1);
NumBits = 1+ceil(log(HowMany)/log(Base));
VetBase = Base.^(-(1:NumBits));
WorkVet = zeros(1,NumBits);
for i=1:HowMany
  j=1;
  ok = 0;
  while ok == 0
    WorkVet(j) = WorkVet(j)+1;
    if WorkVet(j) < Base
      ok = 1;
    else
```

```
      WorkVet(j) = 0;
      j = j+1;
    end
  end
  Seq(i) = dot(WorkVet,VetBase);
end
```

Comment: If we generate a Halton sequence with base $m = 2$, then we see that the first few numbers are

$$\left(\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16}, \frac{5}{16}, \frac{13}{16}, \dots\right)$$

Plotting these numbers shows that they evenly fill out the interval $(0,1)$. Notice that, these numbers would be a bad choice for a symmetric random walk because every step to the right would be followed by a step to the left.

Multidimensional Normal Generation

Definition: A random vector $X = (X_1, X_2, \dots, X_n)$ is said to have a n-dimensional Normal Distribution with a mean vector $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, and a variance-covariance matrix $\Sigma = (\sigma_{i,j})_{i,j=1,\dots,n}$ if the density function of X is given by

$$\phi(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right)$$

To generate a realization of an n-variate Normal vector, we will follow these steps:

1. Use Cholesky decomposition of matrix Σ to write $\Sigma = \mathrm{L}\,\mathrm{L}'$, where L is an $n \times n$ lower-diagonal matrix, and L' is its transpose matrix.
2. Generate a vector $Z = (Z_1, Z_2, \dots, Z_n)$ of independent standard normally distributed variates,
3. Take $X = \mu + \mathrm{L}\,Z$. Then, for this vector, $X \sim N(\mu, \Sigma)$.

To prove the statement of Step 3, note that

(a) $X = \mu +$ L Z is a linear combination of normals, so it is a normal variate,

(b) $EX = \mu +$ L $\mathbb{E}$Z $= \mu$,

(c) $Var(X) = Var(\mu +$ L Z$) = Var($ L Z$) =$ L Var(ZZ′) L′ $=$ L L′ $= \Sigma$.

Since normal variates are determined by their first two moments, then the statements of normality are proved.

Comment:

When $n = 2$, then a Bivariate Normal $X = (X_1, X_2)$ is generated as follows:

Assume $\Sigma = \begin{pmatrix} {\sigma_1}^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & {\sigma_2}^2 \end{pmatrix}$, then we can take L $= \begin{pmatrix} \sigma_1 & 0 \\ \sigma_2\rho & \sqrt{1-\rho^2}\sigma_2 \end{pmatrix}$, and the generation of the two normal variates is given by the following algorithm:

$$\begin{cases} X_1 = \mu_1 + \sigma_1 Z_1 \\ X_2 = \mu_2 + \sigma_2\rho Z_1 + \sigma_2\sqrt{1-\rho^2} Z_2 \end{cases}$$

It is easy to see that the means and variance-covariances of the pair are as desired.

Suggested Exercise: Derive the matrix L for *n=3* case.

Simulation of Stochastic Processes

In this section we will review the mathematical foundations that for the valuation of derivative securities, the Black-Scholes pricing framework in particular. We will cover the basics of the stochastic processes, the popular models of stock prices and their properties, by focusing only on the main concepts and providing intuitions behind them, rather than attempting to give a full and rigorous coverage of the material.

In most models of Asset Pricing, the dynamics of the underlying security follow continuous-time stochastic processes. To model the dynamics of asset prices we use Stochastic Differential Equations (SDEs). In the Black-Scholes framework, the price of the underlying asset under consideration, S_t at time t, follows a Geometric Brownian Motion (GBM). That is, the price dynamics (of a non-dividend paying stock) is given by the following SDE:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

where μ is the drift, and σ is the volatility of the stock price movements. Then, the instantaneous return of the stock is

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t$$

and it is given by two terms: the deterministic term, μdt, and the stochastic term - σdW_t. Here, W_t is a Standard Brownian Motion process, thus, $dW_t \sim N(0, dt)$ and, therefore, $\mu dt + \sigma dW_t \sim N(\mu dt, \sigma^2 dt)$. That is, based on the above dynamics, the stock price return is normally distributed.

In order to better understand the dynamics of the price-process above, study its properties, and the Black-Scholes option pricing formula, we will study SDEs in a slightly more general context.

The Brownian Motion Process

A stochastic process $\{X_t : 0 \leq t \leq T\}$ defined on a probability space $(\Omega, \Lambda, \mathrm{P})$ is called a Brownian Motion Process with a drift μ and variance σ^2, if it satisfies the following properties:

1. $X_0 = 0$,
2. X_t is almost surely continuous in time t,
3. $X_t - X_s$ is independent of $X_s - X_u$ for any $0 \leq u < s < t \leq T$.

4. $X_t - X_s \sim N(\mu(t-s), \sigma^2(t-s))$

A special case of this is the Standard Brownian Motion Process, in which $\mu = 0, \sigma^2 = 1$. This process is also referred to as the Wiener Process, and W_t is used for its notation.

Some properties of Wiener Processes

The Wiener Process $\{W_t: 0 \leq t \leq T\}$ has the following properties:

1. $E\ W_t = 0$ and $Var\ W_t = t$ for any $0 \leq t \leq T$,
2. $\lim_{t\to\infty} \frac{W_t}{t} = 0$
3. $E\ (W_t - W_s|\Lambda_s) = 0$, where $\Lambda_s = \sigma\{W_u,\ u \leq s\}$ is the σ −algebra (or the information set) generated by the Wiener process up to time s.
4. Define the following stopping time: $T_a = \inf\{t \geq 0, W_t = a\}$. Then, for $a > 0$, $P(T_a < t) = 2\ P(W_t > a)$ (this follows from the reflection principle).
5. Compute $P(T_a < \infty)$ and ET_a.

Brownian Bridge Process

A useful modification of the Standard Brownian Motion Process yields a process that is "tied" on both ends of the time interval and behaves like the Brownian Motion Process in the interval. Assume $\{W_t: 0 \leq t \leq T\}$ is a Wiener Process. Define another process B_t on $0 \leq t \leq T$ as follows.

$$B_t = a + W_t - \frac{t}{T}[W_t - b + a]$$

Then, B_t is a Brownian Bridge Process for which $B_0 = a, B_T = b$. This process (by its construction) is a Gaussian Process and is determined by its first and second moments.

Simulation of the Wiener process

Assume $\{W_t: 0 \leq t \leq T\}$ is a Wiener Process. Since we know that $W_s \sim N(0, s)$, then for a Standard Normally distributed random variable $Z \sim N(0,1)$, $\sqrt{T}Z \sim W_T$. That is, W_T has the same distribution as $\sqrt{T}Z$. Therefore, the generation of W_T is straightforward since we have methods to generate Z.

Method 1. Generate $Z \sim N(0,1)$, by one of the two methods learned. Then take $\sqrt{T}Z = W_T$.

Method 2. Notice, that we can write

$$W_T = W_0 + \left(W_{\frac{T}{n}} - W_0\right) + \left(W_{\frac{2T}{n}} - W_{\frac{T}{n}}\right) + \cdots + \left(W_T - W_{\frac{(n-1)T}{n}}\right)$$

Since all the consecutive increments of a Wiener Process above are independent and normally distributed $\left(W_{\frac{kT}{n}} - W_{\frac{(k-1)T}{n}}\right) \sim N\left(0, \frac{T}{n}\right)$, for any $k = 1, \ldots, n$, then we can generate the entire path of the process (actually, it is the discretized version of the process at n pre-specified times).

STEP 1. Generate $\{Z_i\}_{i=1}^{n} \sim iid\ N(0,1)$,

STEP 2. Take $W_T = \sqrt{T/n}\ \sum_{i=1}^{n} Z_i$.

The first method is much easier and computationally not-expensive to implement than the second method for generation of realizations of a Wiener Process at a fixed time T. However, the second method will prove to be more useful as it allows us to generate entire paths of the Wiener Process, not only its terminal value at time T.

Pricing Options – the First Steps

In the Black-Scholes framework, the price of the asset under consideration, S_t at time t, follows a Geometric Brownian Motion (GBM).

That is, the price dynamics (of a non-dividend paying stock) is given by the following SDE:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

In the Risk-Neutral world (or under the risk-neutral measure), the price dynamics is as follows:

$$dS_t = rS_t dt + \sigma S_t dW_t$$

The explicit formula for S_T is given by $S_T = S_0\, e^{\left(\left(r-\frac{\sigma^2}{2}\right)T+\sigma W_T\right)}$, which shows that S_T is Log-Normally distributed (why?).

The price of a European Call option (under the risk neutral measure) is given by

$$c = e^{-(rT)}\mathbb{E}^*(S_T - X)^+ = e^{-(rT)}\mathbb{E}^*\left(S_0\, e^{\left(\left(r-\frac{\sigma^2}{2}\right)T+\sigma W_T\right)} - X\right)^+ = \mathbb{E}^* f(W_T)$$

To estimate c by simulation, we will use the LLN and Monte Carlo simulations to write

$$c = \mathbb{E}^* f(W_T) \approx \frac{1}{n}\sum_{i=1}^{n} f(W_T^i),$$

where W_T^i is the i^{th} realization of the random variable W_T.

The Black-Scholes Formula

In the risk-neutral world, the price of a security (a call option in this case) is simply the discounted (at a risk-free rate) payoff of the security. In case of a European call option, the price c will be given by

$$c = e^{-rT}\mathbb{E}^*(S_T - X)^+$$

In case of a European put option, the price p would be computed as follows:

$$p = e^{-rT}\mathbb{E}^*(X - S_T)^+$$

In both cases, under the risk-neutral measure, the underlying stock-price is given by

$$S_T = S_0\, e^{\left(\left(r-\frac{\sigma^2}{2}\right)T+\sigma W_T\right)}$$

The explicit formulas for prices of European call and put options are given by

$$c = S_0\ N(d_1) - Xe^{-rT}N(d_2), \text{ and } p = Xe^{-rT}N(-d_2) - S_0\ N(-d_1)$$

where

$$d_1 = \frac{\ln\left(\frac{S_0}{X}\right)+\left(r+\frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}},\ d_2 = d_1 - \sigma\sqrt{T}$$

There is also a relationship between the prices of European call and put options on the same stock, with the same strike price and the same maturity, called the Put-Call Parity and given by:

$$c + Xe^{-rT} = p + S_0$$

Comments:

1. When the stock pays dividends at a continuous rate δ, then the above pricing formulas hold true with a minor modification: replace S_0 by $\bar{S}_0 = S_0\, e^{-\delta T}$.
2. When the stock pays discrete dividends , then the above pricing formulas hold true with a minor modification: replace S_0 by:

   $\bar{S}_0 = S_0 - PV\ of\ Dividends\ paid\ during\ the\ life\ of\ the\ option.$

The Greeks of the Black-Scholes Formula

To manage the risk of options and be able to replicate their payoffs, one needs to know the sensitivities of options prices with respect to various parameters of the model – the Greeks.
In the table below, we provide the explicit formulas for European call and put options' greeks.

Table 3**:** ***The Formulas for European Call/Put Option Greeks.***

Greek	**Sensitivity**	**Call option**	**Put option**
Delta	$\frac{\partial Price}{\partial S_0}$	$N(d_1)$	$N(d_1) - 1$
Gamma	$\frac{\partial^2 Price}{\partial {S_0}^2}$	$\frac{1}{S_0 \sigma\sqrt{T}} n(d_1)$	$\frac{1}{S_0 \sigma\sqrt{T}} n(d_1)$
Theta	$\frac{\partial Price}{\partial T}$	$\frac{-S_0 \sigma n(d_1)}{2\sqrt{T}} - rXe^{-rT} N(d_2)$	$\frac{-S_0 \sigma n(d_1)}{2\sqrt{T}} + rXe^{-rT} N(-d_2)$
Vega	$\frac{\partial Price}{\partial \sigma}$	$S_0 \sqrt{T}\, n(d_1)$	$S_0 \sqrt{T}\, n(d_1)$
Rho	$\frac{\partial Price}{\partial r}$	$X\, T\, e^{-rT}\, N(d_2)$	$- X\, T\, e^{-rT}\, N(-d_2)$

where $n(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$, $N(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-u^2/2}\, du$

Estimation of Black- Scholes Greeks by simulations

To estimate the Greeks of option prices, we would use the following method of estimating differentials: Let Δ_c be the delta of a European call option: $\Delta_c = \frac{\partial c}{\partial S_0}$. From the definition of the derivative we know that

$$\Delta_c = \lim_{h \to 0} \frac{C(S_0 + h) - C(S_0)}{h}$$

A natural choice for approximating the derivative is to use a forward-difference scheme:

$$\frac{C(S_0 + \epsilon) - C(S_0)}{\epsilon}$$

This, however, is not the only available method for approximating the delta. We can also approximate delta by a backward-difference

$$\frac{C(S_0) - C(S_0 - \epsilon)}{\epsilon}$$

or, by a central difference:

$$\frac{C(S_0 + \epsilon) - C(S_0 - \epsilon)}{2\epsilon}.$$

Which scheme is better?

A Taylor expansion of the option price about the point (S_0) shows that

$$C(S_0 + \epsilon) = C(S_0) + \epsilon \frac{\partial C(S_0)}{\partial S_0} + \frac{1}{2}\epsilon^2 \frac{\partial^2 C(S_0)}{\partial {S_0}^2} + O(\epsilon^3).$$

$O(\epsilon^3)$, called "big O notation", means that the remainder of this expansion is smaller in absolute value than $|\epsilon^3|$ times a constant. Similarly, for the backward difference, we can write,

$$C(S_0 - \epsilon) = C(S_0) - \epsilon \frac{\partial C(S_0)}{\partial S_0} + \frac{1}{2}\epsilon^2 \frac{\partial^2 C(S_0)}{\partial {S_0}^2} + O(\epsilon^3).$$

Thus, we conclude that

$$\frac{\partial c}{\partial S_0} = \frac{C(S_0 + \epsilon) - C(S_0 - \epsilon)}{2\epsilon} + O(\epsilon^2)$$

The error of the central difference scheme is therefore of size $O(\epsilon^2)$. From the above equations, it follows immediately that the error of the forward and backward difference schemes are of size $O(\epsilon)$. Therefore, the central difference scheme provides greater accuracy, but there are reasons (e.g. stability) for using the forward or the backward difference. In some situations, when where S is close to zero, it might not even be possible to use the central difference scheme because we can't estimate $C(S_0 - \epsilon)$. It is possible to derive forward and backward schemes with greater accuracy by using three points instead of two (see the problems below).

We can estimate the other Greeks by simulation in a very similar fashion:

Perturb the initial parameter under consideration, and estimate the price of the option under the new parameter. Then, approximate the differential by a finite difference. For example, to estimate the Vega, we can use the forward difference method as follows:

$$\frac{C(\sigma + \epsilon) - C(\sigma)}{\epsilon}$$

Problems.

1. Show that the delta of a stock option can also be approximated by

$$\frac{-3C(S_0) + 4C(S_0 + \epsilon) - C(S_0 + 2\epsilon)}{2\epsilon}.$$

   Show that this approximation is of order $O(\epsilon^2)$.

2. Derive a backward scheme which is based on the points $(S_0), (S_0 - \epsilon)$ and $(S_0 - 2\epsilon)$ and is of the order $O(\epsilon^2)$.
3. Approximate the Gamma and the Theta by using the forward difference method.

The Black-Scholes PDE

The following partial differential equation is the Black-Scholes PDE, which gives the price a European call option:

$$\frac{\partial c}{\partial t} + rS\frac{\partial c}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 c}{\partial S^2} - rc = 0$$

The PDE is the same whether one is pricing call or put options on the same underlying. It is the terminal conditions that differentiate European call or put option prices.

Simulation of paths of Stochastic Differential Equation (SDE)

Consider a general (not the most general) one-dimensional Ito's Stochastic Process given by:

$$X_t = x_0 + \int_0^t a(X_s)ds + \int_0^t b(X_s)dW_s$$

The SDE for of the above integral equation is given by the following:

$$dX_t = a(X_t)dt + b(X_t)dW_t, \text{ and } X_0 = x_0.$$

First, we will provide a result that will allow us to write an SDE for a function of an Ito's stochastic process, for which the SDE is known.

Ito's Formula

Suppose the stochastic process $\{X_t,\ t \geq 0\}$ is an Ito Process. That is, it can be written in the following integral form:

$$X_t = x_0 + \int_0^t a(X_s, s)ds + \int_0^t b(X_s\, s)dW_s$$

The SDE of the above integral equation is given by

$$dX_t = a(X_t, t)dt + b(X_t, t)dW_t, and\ X_0 = x_0.$$

We would like to be able to write the dynamics of a function $f(X_t, t)$ of the process $\{X\_t,\ t \geq 0\}$ and time in an SDE form. The next result will help to answer this question.

Lemma (Itô). Assume the function $f(\cdot,\cdot) \in C^2(R \times R^+)$ and the process $\{X_t,\ t \geq 0\}$ satisfies the SDE:

$$dX_t = a(X_t, t)dt + b(X_t, t)dW_t, and\ X_0 = x_0.$$

Then, the SDE for $\{f(X_t, t),\ t \geq 0\}$ will be given by

$$df(X_t,t) = \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial x}dX + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(dX)^2$$

$$= \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}a(X_t,t) + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}b(X_t,t)^2\right)dt + \frac{\partial f}{\partial x}\, b(X_t,t)dW_t$$

Comment: In Ito's calculus, we will use the following products when dealing with terms like $(dX_t)^2$:

$$dt\, dt = dt\, dW_t = dW_t dt = 0, and\ dW_t\, dW_t = dt$$

Example: $f(W_t,t) = \frac{1}{2}W^2{}_t$. Then

$$df(W_t,t) = d\left(\frac{1}{2}W^2{}_t\right) = 0dt + \frac{1}{2}2W_t dW_t + \frac{1}{2}(\frac{1}{2}\ 2\ 1)dt = W_t dW_t + \frac{1}{2}dt.$$

Integrating both sides will result in

$$\int_0^s W_t dW_t = \frac{1}{2}W_s^2 - \frac{1}{2}s$$

Higher Dimensional Ito's Formula

Assume the n-dimensional stochastic vector process $\{X_t,\ t \geq 0\}$ satisfies

$$dX_t = a(X_t,t)dt + b(X_t,t)dW_t, and\ X_0 = x_0.$$

where

$$X_t = \begin{pmatrix} X_t^1 \\ X_t^2 \\ . \\ . \\ X_t^n \end{pmatrix}, a(X_t,t) = \begin{pmatrix} a^1(X_t,t) \\ a^2(X_t,t) \\ . \\ . \\ a^n(X_t,t) \end{pmatrix},$$

$$b(X_t,t)=\begin{pmatrix} b^{11}(X_t,t),\dots,b^{1m}(X_t,t) \\ b^{21}(X_t,t),\dots,b^{2m}(X_t,t) \\ . \\ . \\ b^{n1}(X_t,t),\dots,b^{nm}(X_t,t) \end{pmatrix},\ dW_t=\begin{pmatrix} dW_t^1 \\ dW_t^2 \\ . \\ . \\ dW_t^m \end{pmatrix}$$

Let $f(t,x)=\big(f_1(t,x),\dots,f_k(t,x)\big)$ be $C^2(R^+\times R^n)\to R^k$.

Then, the SDE for $\{f(t,X_t),\ t\geq 0\}$ will be given by

$$df_i(t,X_t) \quad =\frac{\partial f_i}{\partial t}dt+\sum_{j=1,..,n}\frac{\partial f_i}{\partial x_j}dX_j+\frac{1}{2}\sum_{j,l=1,..,n}\frac{\partial^2 f_i}{\partial x_j\partial x_l}\big(dX_j\ dX_l\big)$$

where, we use the following properties

$$dt\ dt=dt\ dW_t^i=dW_t^i dt=0, \qquad \text{and } dW_t^i\ dW_t^i=dt,$$

$$dW_t^i\ dW_t^j=0 \text{ for any } i,j=1,\dots,n, \text{and } i\neq j.$$

Example: Consider the following 2-dimensional example.

$$dX_t=(a-X_t)dt+\sqrt{X_t}dW_t^1+bY_t dW_t^2$$

$$dY_t=(cX_tY_t)dt+(Y_t+1)dW_t^1+edW_t^2$$

and $X_0=1,\ Y_0=1/2$.

Define $f(t,x,y)=\big(f_1(t,x,y),f_2(t,x,y),f_3(t,x,y)\big)$ to be $C^2(R^+\times R^2)\to R^3$ as follows:

$$f_1(t,x,y)=txy,\ f_2(t,x,y)=x+y,\ f_3(t,x,y)=e^{x+y}.$$

Then,

$$df_1(t,X_t,Y_t)=X_tY_tdt+tX_tdY_t+tY_tdX_t+\big(t[\ \big(\sqrt{X_t}\big)(Y_t+1)+beY_t\ ]\big)d$$

$$=\big(X_tY_t+ctX^2{}_tY_t+tY_t(a-X_t)+\ \big(t[\ \big(\sqrt{X_t}\big)(Y_t+1)+beY_t\ ]\big)\big)dt$$

$$+\Big((tX_t(Y_t+1)+tY_t)\sqrt{X_t}\Big)\ dW_t^1+\ (tX_te+tbY^2{}_t)dW_t^2.$$

Similarly we can express $df_2(t, X_t, Y_t)$ and $df_3(t, X_t, Y_t)$ in an SDE form.

DISCRETIZATION OF SDEs

Consider a general (not the most general) Ito's Stochastic Process given by:

$$X_t = x_0 + \int_0^t a(X_s)ds + \int_0^t b(X_s)dW_s$$

The SDE for of the above integral equation is given by the following:

$$dX_t = a(X_t)dt + b(X_t)dW_t, \text{and } X_0 = x_0.$$

The above integral equation or the SDE do not always have explicit solutions, so there is a need for numerical algorithms for solving these equations numerically. The basic idea for numerically solving equations similar to the integral equation or the SDE above is explained below.

Discretize the $[0, T]$ interval by the partition $0 = t_0 < t_1 < \cdots < t_N = T$. Denote $\Delta = \frac{T}{N}$. There are many ways to discretize the SDE above, of which we will discuss just a few. We will consider the Euler's and Milshtein's discretization schemes below.

First, we will approximate the solutions of the above SDE by considering discrete time approximations that are derived from the stochastic Taylor expansion.

Assume the SDE for X_t is given by:

$$dX_t = a(X_t)dt + b(X_t)dW_t, \text{and } X_0 = x_0.$$

The SDE can be written in an equivalent integral form:

$$X_t = x_0 + \int_0^t a(X_s)ds + \int_0^t b(X_s)dW_s$$

Now, applying the Ito's formula on $a(X_t)$ and on $b(X_t)$, we will get SDEs for $a(X_t)$ and on $b(X_t)$:

$$\begin{aligned} da(X_t) &= a'(X_t)dX_t + \frac{1}{2}a''(X_t)(dX_t)^2 \\ &= \left(a'(X_t)a(X_t) + \frac{1}{2}a''(X_t)b(X_t)^2\right)dt + a'(X_t)\, b(X_t)dW_t \\ db(X_t) &= b'(X_t)dX_t + \frac{1}{2}b''(X_t)(dX_t)^2 \\ &= \left(b'(X_t)a(X_t) + \frac{1}{2}b''(X_t)b(X_t)^2\right)dt + b'(X_t)\, b(X_t)dW_t \end{aligned}$$

These SDEs can be written as integral equations as follows:

$$a(X_s) = a(x_0) + \int_0^s \left(a'(X_u)a(X_u) + \frac{1}{2}a''(X_u)b(X_u)^2\right)du + \int_0^s a'(X_u)b(X_u)dW_u$$

$$b(X_s) = b(x_0) + \int_0^s \left(b'(X_u)a(X_u) + \frac{1}{2}b''(X_u)b(X_u)^2\right)du + \int_0^s b'(X_u)b(X_u)dW_u$$

Replace $a(X_s)$and $b(X_s)$ in the original integral equation, $X_t = x_0 + \int_0^t a(X_s)ds +$ $\int_0^t b(X_s)dW_s$, by their expressions derived above to get the following integral equation:

$$X_t = x_0 +$$

$$\int_0^t \left\{a(x_0) + \int_0^s \left(a'(X_u)a(X_u) + \tfrac{1}{2}a''(X_u)b(X_u)^2\right)du + \int_0^s a'(X_u)b(X_u)dW_u\right\} ds +$$

$$\int_0^t \left\{b(x_0) + \int_0^s \left(b'(X_u)a(X_u) + \tfrac{1}{2}b''(X_u)b(X_u)^2\right)du + \int_0^s b'(X_u)b(X_u)dW_u\right\} dW_s$$

This will lead to the Ito-Taylor expansion of X_t.

All Numerical algorithms for generating paths of the process X_t are based on the above formula. We will consider a few of those algorithms.

The Euler's scheme

We can use the above integral equation and approximate the X_t process by taking a few terms on the right-hand side, and ignoring the rest of the right-side.

$$X_t = x_0 + \int_0^t \{a(x_0)\}\, ds + \int_0^t \{b(x_0)\}\, dW_s \; +$$

$$\int_0^t \left\{ \int_0^s \left(a'(X_u)a(X_u) + \frac{1}{2}a''(X_u)b(X_u)^2 \right) du + \int_0^s a'(X_u)b(X_u)dW_u \right\} ds +$$

$$\int_0^t \left\{ \int_0^s \left(b'(X_u)a(X_u) + \frac{1}{2}b''(X_u)b(X_u)^2 \right) du + \int_0^s b'(X_u)b(X_u)dW_u \right\} dW_s$$

If we only take the first three terms of the right hand side (which were highlighted in red color), then we will get this approximate discrete version of the above SDE:

$$\tilde{X}_t = \tilde{X}_0 + a(\tilde{X}_0)t + b(\tilde{X}_0)W_t$$

This approximation (or discrete scheme) can be applied to all time intervals to yield the following scheme:

The general term can be written by

$$\tilde{X}_{t_{k+1}} - \tilde{X}_{t_k} = a(\tilde{X}_{t_k})\Delta + b(\tilde{X}_{t_k})\Delta W_{t_k}, \text{and } \tilde{X}_0 = x_0.$$

Here $\Delta W_{t_k} = W_{t_{k+1}} - W_{t_k}$.

We know that, for a Wiener process W_t, $\Delta W_{t_k}, k = 0, \dots, N-1$ are independent increments and are normally distributed with the same distribution: $\Delta W_{t_k} \sim N(0, \Delta)$.

Theorem. Under certain regularity conditions (Lipschitz and Bounded Growth), the discrete version of the simulated $\tilde{X}_N$ converges in a strong sense to the continuous-time version of X_T as $N \to \infty$, with order of convergence ½:

$$\mathbb{E}\left(\left|\tilde{X}_N - X_T\right|\right) \leq K\left(\frac{T}{N}\right)^{1/2}$$

That is, the convergence rate of the Euler scheme is $N^{-1/2}$.

There are other schemes that result in better convergence rates and we will consider one of them below – the Milshtein's scheme.

How do we simulate the path of the stochastic process $\{X_t, 0 \leq t \leq T\}$ using Euler's scheme?

In case of the Euler scheme we can use the following algorithm:

STEP 1: Set $\tilde{X}_{t_0} = x_0$,

STEP 2: Generate a sequence of $Z_1, Z_2, \dots, Z_N \sim iid\ N(0,1)$,

STEP 3: Define $\tilde{X}_{t_{k+1}} = \tilde{X}_{t_k} + a\left(\tilde{X}_{t_k}\right)\Delta + b\left(\tilde{X}_{t_k}\right)\sqrt{\Delta}\, Z_{k+1}$ for $k = 0,1, \dots, N-1$.

Thus, we have a simulated a path $\{\tilde{X}_{t_k}, k = 0,1, \dots, N\}$ of the stochastic process $\{X_t, 0 \leq t \leq T\}$ at discrete times $0 = t_0 < t_1 < \cdots < t_N = T$.

Comment: The Euler scheme evaluates the process at discrete times: $t_0 < t_1 < \cdots < t_N$. If needed, one can also approximate the values of the process at times that are in between the discretization times, say at time t so that $t_k < t < t_{k+1}$. To do this, one can use a simple linear interpolation to write:

$$\tilde{X}_t = \tilde{X}_{t_k} + \frac{t - t_k}{t_{k+1} - t_k}\left(\tilde{X}_{t_{k+1}} - \tilde{X}_{t_k}\right)$$

The Milshtein's Scheme

Another discretized version of the above SDE that uses more terms in the Ito-Taylor expansion is given by the Milshtein's scheme:

$$\tilde{X}_{t_{k+1}} - \tilde{X}_{t_k} = a(\tilde{X}_{t_k})\Delta + b(\tilde{X}_{t_k})\Delta W_{t_k} + \frac{1}{2}b(\tilde{X}_{t_k})b'(\tilde{X}_{t_k})\{\Delta W_{t_k}{}^2 - \Delta\}, \text{and } X_0 = x_0.$$

Here, $\Delta W_{t_k} = W_{t_{k+1}} - W_{t_k}$.

We know that the X_t can be written as follows:

$$X_t = x_0 + \int_0^t \{a(x_0)\}\, ds + \int_0^t \{b(x_0)\}\, dW_s \; +$$

$$\int_0^t \left\{ \int_0^s \left(a'(X_u)a(X_u) + \frac{1}{2}a''(X_u)b(X_u)^2 \right) du + \int_0^s a'(X_u)b(X_u)dW_u \right\} ds +$$

$$\int_0^t \left\{ \int_0^s \left(b'(X_u)a(X_u) + \frac{1}{2}b''(X_u)b(X_u)^2 \right) du + \int_0^s b'(X_u)b(X_u)dW_u \right\} dW_s$$

We will rewrite the above equation as follows:

$$X_t = x_0 + \int_0^t \{a(x_0)\}\, ds + \int_0^t \{b(x_0)\}\, dW_s \; + \int_0^t \int_0^s b'(X_u)b(X_u)dW_u\, dW_s +$$

$$\int_0^t \left\{ \int_0^s \left(a'(X_u)a(X_u) + \frac{1}{2}a''(X_u)b(X_u)^2 \right) du + \int_0^s a'(X_u)b(X_u)dW_u \right\} ds +$$

$$\int_0^t \left\{ \int_0^s \left(b'(X_u)a(X_u) + \frac{1}{2}b''(X_u)b(X_u)^2 \right) du \right\} dW_s$$

We will use the first three terms and only a part of the forth term of the right-side of the above expression to derive the Milshtein's scheme. To derive the scheme, we apply the Ito's lemma on the function $f(X_u) = b'(X_u)b(X_u)$ and replace the integrand in the integral $\int_0^t \int_0^s b'(X_u)b(X_u)dW_u\, dW_s$ above by that expression. Then, we will take only the first term of that integral's expansion and add to the other terms that determined the Euler's scheme to get the following:

$$X_t = x_0 + \int_0^t \{a(x_0)\}\, ds + \int_0^t \{b(x_0)\}\, dW_s + b'(x_0)b(x_0) \int_0^t \int_0^s dW_u\, dW_s +$$

(Higher Order Terms for $b'(X_u)b(X_u)$ expansion)+

$$\int_0^t \left\{ \int_0^s \left(a'(X_u)a(X_u) + \frac{1}{2}a''(X_u)b(X_u)^2 \right) du + \int_0^s a'(X_u)b(X_u)dW_u \right\} ds +$$

$$\int_0^t \left\{ \int_0^s \left(b'(X_u)a(X_u) + \frac{1}{2}b''(X_u)b(X_u)^2 \right) du \right\} dW_s$$

Use the fact that, $\int_s^T W_t dW_t = \frac{1}{2}(W_T - W_s)^2 - \frac{1}{2}(T - S)$, to get the Milshtein's scheme:

$$\tilde{X}_{t_{k+1}} - \tilde{X}_{t_k} = a(\tilde{X}_{t_k})\Delta + b(\tilde{X}_{t_k})\Delta W_{t_k} + \frac{1}{2}b(\tilde{X}_{t_k})b'(\tilde{X}_{t_k})\{(\Delta W_{t_k})^2 - \Delta\}, \text{and } X_0 = x_0$$

The following result justifies the use of the Milshtein's scheme in approximating SDE solutions.

Theorem: Under certain regularity conditions (Lipschitz and Bounded Growth), the discrete version of the simulated process $\tilde{X}_N$ converges in a strong sense to the continuous-time version of X_T as $N \to \infty$, with order 1:

$$\mathbb{E}(|\tilde{X}_N - X_T|) \leq K \left(\frac{T}{N}\right)^1$$

That is, the rate of strong convergence of the Milshtein's scheme is N^{-1}. This is a significant improvement in the convergence rate, though it comes at the expense of adding an extra term. In the end, it is a numerical exercise to compare the cost and benefits of adding that extra term and using the Milshtein's scheme, versus the Euler's scheme.

The Heston Model

Consider the following 2-factor model for stock prices with stochastic volatility:

$$\begin{cases} dS_t = rS_t dt + \sqrt{V_t}\, S_t dW_t^1 \\ dV_t = \alpha(\beta - V_t)dt + \sigma\sqrt{V_t}\, dW_t^2 \end{cases}$$

where the Brownian Motion processes above are correlated: $dW_t^1 dW_t^2 = \rho dt$, where the correlation ρ is a constant in $[-1,1]$. This model is referred to as the Heston Model.

Many practical applications of models with Heston-dynamics involve the pricing and hedging of path-dependent securities, which, in turn, nearly always require uses of Monte Carlo methods.

Remark: Taking $V_0 > 0$ and $2\alpha\beta \geq \sigma^2$ guarantees that the process for V_t can never reach zero. In typical applications, however, $2\alpha\beta$ is often significantly below σ^2, so the chance of V_t hitting zero is significant. Also, in simulations often one faces the case of V_t **turning negative**, despite the choice of parameters that satisfy $2\alpha\beta \geq \sigma^2$.

To price securities one often needs to solve the above model numerically, using Monte Carlo simulations. Therefore a discretization scheme is used to simulate the dynamics of the price and volatility processes above. For demonstration purposes, we will use the Euler discretization method: Let's consider the Euler discretization scheme:

$$\hat{S}_{k+1} - \hat{S}_k = r\hat{S}_k \, \Delta + \sqrt{\hat{V}_k} \, \hat{S}_k \, \sqrt{\Delta} \, Z_{k+1}^1$$

$$\hat{V}_{k+1} - \hat{V}_k = \alpha(\beta - \hat{V}_k) \, \Delta + \sigma \sqrt{\hat{V}_k} \, \sqrt{\Delta} \, Z_{k+1}^2$$

$$\text{for } k = 0,1, \dots, N-1$$

Here, $\hat{V}_0 = V_0$ and $\hat{S}_0 = S_0$ are given. Also, Z_{k+1}^1 and Z_{k+1}^2 are standard normally distributed random variables with correlation ρ.

Since this is a finite discretization of a continuous process, it is possible that V becomes negative with non-zero probability, which in turn would make computation of $\sqrt{\hat{V}_k}$ impossible, and cause the scheme to fail. It could happen due *discretization errors* of the discrete scheme.

To get around this problem, several remedies have been proposed in the literature; see the paper by Lord, R., R. Koekkoek and D. van Dijk (2006), "*A Comparison of biased simulation schemes for stochastic volatility models*" for a review of various "fixes" of the problem.

In order to handle negative values of V, we need to modify the above scheme to include methods of eliminating negative values for subsequent iterations of the volatility path. Thus we introduce three new schemes, summarized below, by using three functions as defined below.

$$\hat{S}_{k+1} - \hat{S}_k = r\hat{S}_k\,\Delta + \sqrt{f_3(\hat{V}_k)}\,\hat{S}_k\,\sqrt{\Delta}\,Z_{k+1}^1$$

$$\hat{V}_{k+1} - f_1(\hat{V}_k) = \alpha(\beta - f_2(\hat{V}_k))\,\Delta + \sigma\sqrt{f_3(\hat{V}_k)}\;\sqrt{\Delta}\,Z_{k+1}^2$$

$$\text{for } k = 0,1,\dots,N-1$$

Here, $\hat{V}_0 = V_0$ and $\hat{S}_0 = S_0$ are given.

Scheme	$\boldsymbol{f_1}$	$\boldsymbol{f_2}$	$\boldsymbol{f_3}$
Reflection	$\lvert x\rvert$	$\lvert x\rvert$	$\lvert x\rvert$
Partial Truncation	x	x	x^+
Full Truncation	x	x^+	x^+

The existing literature suggests that the **Full Truncation method is the "best"**. Here is demonstration of the application of the full truncation method to the Heston Model of Stock Price with Stochastic Volatility:

$$\hat{S}_{k+1} - \hat{S}_k = r\hat{S}_k\,\Delta + \sqrt{\hat{V}_k^{\,+}}\,\hat{S}_k\,\sqrt{\Delta}\,Z_{k+1}^1$$

$$\hat{V}_{k+1} - \hat{V}_k = \alpha(\beta - \hat{V}_k^{+})\,\Delta + \sigma\sqrt{\hat{V}_k^{+}}\,\sqrt{\Delta}\,Z_{k+1}^2$$

for $k = 0,1,\dots,N-1$, where $\hat{V}_0 = V_0$ and $\hat{S}_0 = S_0$ are given, $\hat{V}_k^{+} = \max(0, \hat{V}_k)$.

Remark:

The main characteristic of this scheme is that the process for V is allowed to go below zero, at which point the process for V becomes deterministic with an upward drift of $\alpha\beta$.

Numerical Computation of $N(\cdot)$

The CDF of a standard normal distribution does not have a closed-form formula that can be used in computing the prices of options. Thus, it needs to be estimated numerically. Consider the following approximation:

$$N(x) = \begin{cases} 1 - \frac{1}{2}(1 + d_1x + d_2x^2 + d_3x^3 + d_4x^4 + d_5x^5 + d_6x^6)^{-16}, & if\ x \geq 0 \\ 1 - N(-x), & if\ x < 0 \end{cases}$$

Then, with the following choice of the parameters the method has an accuracy of 10^{-7}.

$$d_1 = 0.0498673470, \quad d_2 = 0.0211410061, \quad d_3 = 0.0032776263,$$

$$d_4 = 0.0000380036,\ d_5 = 0.0000488906,\ d_6 = 0.0000053830.$$

Exercises

1. Let S_t be a Geometric Brownian Motion process: .

$$S_t = S_0 \; e^{\left(\sigma W_t + \left(r - \frac{\sigma^2}{2}\right)t\right)}$$

where $r = 0.04, \sigma = 0.2, S_0 = 88$, and W_t is a Standard Brownian Motion process (Wiener process).

   (a) Estimate the price c of a European Call option on the stock with $T = 10, X = 100$ *by using Monte Carlo simulation*.

   (b) Now use the variance reduction techniques to compute the price in part (a) again. Did the accuracy improve? You may compute the exact value of the option c by the Black-Scholes formula. Now estimate c by crude Monte Carlo simulation, and then, by using different variance reduction techniques to see if there is an improvement in convergence.

2. Simulate 4 paths of $S_t \; for \; 0 \leq t \leq 10$ (defined in the Problem 4) by dividing up the interval [0, 10] into 1,000 equal parts. Then, for each integer number n from 1 to 10, compute ES_n. Plot all of this in one graph.

3. Evaluate the following expected values and probabilities:

$$E\left(X_2^{\frac{1}{3}}\right), \qquad E(Y_3), \qquad E\big(X_2 Y_2 \, \mathbf{1}(X_2 > 1)\big), \qquad P(Y_2 > 5).$$

where the Ito's processes X and Y evolve according to the following SDEs:

$$dX_t = \left(\frac{1}{5} - \frac{1}{2}X_t\right)dt + \frac{2}{3}dW_t,\ X_0 = 1,$$

$$dY_t = \left(\left(\frac{2}{1+t}\right)Y_t + \frac{1+t^3}{3}\right)dt + \frac{1+t^3}{3}dZ_t,\ Y_0 = \frac{3}{4}$$

$W\ and\ Z$ are independent Wiener processes, and $\mathbf{1}(X_2 > 1) = 1\ \ if\ X_2 > 1, and\ 0\ if\ X_2 \leq 1$.

4. Estimate the following expected values and compare:

$$E(1+X_3)^{1/3},\ and\ \ E(1+Y_3)^{1/3}$$

where

$$dX_t = \frac{1}{4}X_t dt + \frac{1}{3}X_t dW_t - \frac{3}{4}X_t dZ_t,\ \ X_0 = 1$$

$$Y_t = e^{-0.08t + \frac{1}{3}W_t + -\frac{3}{4}Z_t}$$

and $W\ and\ Z$ are independent Wiener processes.

5. (a) Compute the price of a European Call option via Monte Carlo simulation. Use variance reduction techniques (e.g. antithetic variates). The code should be generic: for any input of the 5 parameters $-\ S_0, T, X, r, \sigma$, the output is the corresponding price of the call option.

(b) Compute the price of a European Call option by using the Black-Scholes formula. (use the approximation of $N(\cdot)$ described in class). The code should be generic: for any input values of the 5 parameters $-\ S_0, T, X, r, \sigma,$ it should compute and return the value of the option price.

(c) Compute the hedging parameters of a Call option (all 5 of the Greeks) and graph them as a function of stock price: S_0, where S_0=20, $X = 20$, $\sigma = 0.25$, $r = 0.04$ and $T = 0.5$ years. Use the range [15:25] for S_0 with a step size 1.

6. Compute the probability (by simulation) that a European *put* option will expire in the money. Use these parameters: $S_0 = 20, X = 20, \sigma = 0.25, r = 0.04$ and $T = 0.5$ years.
7. Compare the pseudorandom sample with the quasi MC sample of $Uniform[0,1]x[0,1]$:
   a) Generate 100 vectors of $Uniform$ $[0,1]x[0,1]$ by using MATLAB (or the software you are using) random number generator.
   b) Generate 100 points of the 2-dimentional Halton sequences, using bases 2 and 7.
   c) Generate 100 points of the 2-dimentional Halton sequences, using bases 2 and 4. (4 is non-prime!).
   d) Draw all on separate graphs and see if there are differences in the three (visual test only).
   e) Use 2-dimensional Halton sequences and compute the following integral:
   (use $N = 10{,}000$. Try different pairs of bases: $(2,4), (2,7), (5,7)$.)

$$\int_0^1 \int_0^1 e^{-xy} \left(\sin(6\pi x) + cos^{\frac{1}{3}} (2\pi y) \right) dxdy$$

8. Consider the following 3-factor model for short term interest rate (Balduzzi et al. model):

$$\begin{cases} dr_t = k(\alpha_t - r_t)dt + \sqrt{v_t} dW_t^1 \\ d\alpha_t = a(b - \alpha_t)dt + cdW_t^2 \\ dv_t = u(x - v_t)dt + z\sqrt{v_t} dW_t^3 \end{cases}$$

where $dW_t^1 dW_t^2 = \rho_1 dt, \quad dW_t^2 dW_t^3 = \rho_2 dt, \; dW_t^3 dW_t^1 = \rho_3 dt$. Compute $E(r_2)$ and $E(e^{-\int_0^2 r_s ds})$ for $\rho_1 = 0.5, \; \rho_2 = 0.2, \; \rho_1 = 0.4$. Choose $k = 0.1027, r_0 = 0.035, \alpha_0 = 0.036, a = 0.089, b = 0.0377, c = 0.05, v_0 = 0.15, u = 0.092, x = 0.18, z = 0.067$.

9. Consider the following 2-factor model for stock prices with stochastic volatility:

$$\begin{cases} dS_t = rS_t dt + \sqrt{V_t}\, S_t dW_t^1 \\ dV_t = \alpha(\beta - V_t)dt + \sigma\sqrt{V_t}\, dW_t^2 \end{cases}$$

where the Brownian Motion processes above are correlated: $dW_t^1 dW_t^2 = \rho dt$, where the correlation ρ is a constant in $[-1,1]$.

Compute the price of a European Call option (via Monte Carlo simulation) that has a Strike price of \$50 and matures in 1 year. Use the following parameters of the model: $\rho = -0.6, r = 0.03,\ S_0 = \$48,\ V_0 = 0.25,\ \sigma = 0.42,\ \alpha = 5.8,\ \beta = 0.12.$

Use the Full Truncation, Partial Truncation and Reflection methods, and compare the efficiencies of the tree methods.

10. X and Y are standard normally distributed random variables. Define another random variable Z as follows: flip a fair coin and if the outcome is a Tale, take $Z = X$, otherwise take $Z = Y$. Describe the distribution of Z. What are the mean and the variance of Z? Use explicit formulas to answer. Use simulations to answer.
11. What is the sum of all the integers from 1 to 100? You have 30 seconds to answer!
12. You hold two European Call options with similar characteristics, but one (the first) matures in 1 year and the other (the second) matures in 3 months. Which will have higher delta? Higher Gamma? Use explicit formulas to answer. Use simulations to answer.
13. Which is more expensive: an ATM European call or an ATM European put on the same stock with the same maturity? Use explicit formulas to answer. Use simulations to answer.

14. Which is higher: the Gamma of an ATM European call or an ATM European put on the same stock with the same maturity? What if they both are 10% ITM or 10% OTM? Use explicit formulas to answer. Use simulations to answer.