Part I
Generating Discrete Random Variables

1 The Inverse Transformation Method

The inverse transformation method is a commonly used method to generate discrete as well as continuous random variables. The general principle is the following theorem.

Theorem 1 *Let $U_1, U_2, ..., U_n$ be a random sample of size n, i.e., i.i.d. random variables, from the uniform distribution $U(0,1)$. Let F be a distribution function and $X_i = F^{-1}(U_i)$. Then $X_1, X_2, ..., X_n$ is a random sample of size n from the distribution F.*

Proof. Simply observe that $P\{X_i \leq x\} = P\{F^{-1}(U_i) \leq x\} = P\{U_i \leq F(x)\} = F(x)$. ■

Next we will discuss some discrete random variables and how to sample from them using the inverse transformation method. Later, we will learn about other methods of generating random variables.

1.1 Discrete uniform random variable

If the space of a random variable X is finite or countable (i.e., if X can take only finitely or countably many values) then X is called a discrete random variable. For a discrete random variable X, we define its probability density function as

$$P(X = x_i) = p_i, i = 1, 2, ...$$

where $\sum p_i = 1$.To generate, or sample, a random value from X, using the inverse transformation method, we first generate a random number u from $U(0,1)$ and set

$$X = \begin{cases} x_1 \text{ if } u < p_1 \\ x_2 \text{ if } p_1 \leq u < p_1 + p_2 \\ \cdots \\ x_i \text{ if } p_1 + ... + p_{i-1} \leq u < p_1 + ... + p_i \\ \cdots \end{cases} \tag{1}$$

If F is the distribution function for X, you can see that the above expression is simply $F^{-1}(u)$.

Next we will discuss specific discrete distributions and how the above equation can be computed efficiently.

1.2 Uniform discrete distribution

Let X be a random variable with space $\{1, 2, ..., n\}$. We say X has a discrete uniform distribution (or, X is a discrete uniform random variable) on the integers $1, 2, ..., n$ if its density is

$$p_i = P\{X = i\} = \frac{1}{n},\ i = 1, ..., n.$$

Recall that for a discrete uniform random variable X on integers $1, 2, ..., n$, we have

$$E[X] = \frac{n+1}{2} \text{ and } Var(X) = \frac{n^2 - 1}{12}.$$

Sampling from a discrete uniform random variable can be done very efficiently. Indeed, from equation (1) we get:

$$\begin{aligned} X &= i \text{ if } \frac{i-1}{n} \leq u < \frac{i}{n},\ i = 1, ..., n \\ &\Rightarrow X = i \text{ if } i - 1 \leq nu < i \\ &\Rightarrow X = i \text{ if } Floor(nu) = i - 1 \\ &\Rightarrow X = Floor(nu) + 1 \end{aligned}$$

So we simply generate a uniform random number u and compute $Floor(nu) + 1$ to get a random integer from the uniform distribution on $1, ..., n$.

1.3 Geometric distribution

Suppose that independent Bernoulli trials, each having a probability of success p, are performed until a success occurs. Let X be the number of trials required. The density function of X is:

$$P(X = i) = p_i = (1 - p)^{i-1} p,\ i = 1, 2, ... \tag{2}$$

We say a random variable X has the geometric distribution if its pdf is given by (2). Recall that the expectation and variance of a geometric random variable with parameter p are

$$E[X] = \frac{1}{p} \text{ and } Var(X) = \frac{1-p}{p^2}.$$

Observe that

$$\begin{aligned} \sum_{j=1}^{i-1} P\{X &= j\} = 1 - P\{X > i - 1\} \\ &= 1 - P\{\text{first } i - 1 \text{ are all failures}\} \\ &= 1 - q^{i-1} \end{aligned}$$

where $j \geq 1$ and $q = 1 - p$ is the failure probability. From (1), we have

$$\begin{aligned} X &= i \text{ if } p_1 + ... + p_{i-1} \leq u < p_1 + ... + p_i \\ &\Leftrightarrow \sum_{j=1}^{i-1} P\{X = j\} \leq u < \sum_{j=1}^{i} P\{X = j\} \\ &\Leftrightarrow 1 - q^{i-1} \leq u < 1 - q^i \\ &\Leftrightarrow q^i < 1 - u \leq q^{i-1} \end{aligned}$$

Taking logarithms, this simplifies to

$$X = i \text{ if } i \log q < \log(1 - u) \leq (i - 1) \log q$$

and simplifying, we get

$$X = i \text{ if } i - 1 \leq \frac{\log(1 - u)}{\log q} < i$$

Therefore

$$X = Floor\left(\frac{\log(1 - u)}{\log q}\right) + 1.$$

If we note that the distribution of $1 - u$ is $U(0, 1)$ if u is $U(0, 1)$, then we obtain the simple procedure: To generate a random number from the geometric distribution X, generate a random number from $U(0, 1)$ and set

$$X = Floor\left(\frac{\log u}{\log q}\right) + 1.$$

1.4 Poisson distribution

We say a random variable X with space $\{0, 1, 2, ...\}$ has the Poisson distribution with parameter λ, and write $X \sim Poisson(\lambda)$ if its density function is

$$P\{X = i\} = p_i = e^{-\lambda} \frac{\lambda^i}{i!},\ i = 0, 1, ...$$

Recall that the mean and variance of X is simply λ, i.e.,

$$E[X] = \lambda \text{ and } Var(X) = \lambda.$$

Observe that p_i can be computed recursively using

$$p_{i+1} = \frac{\lambda}{i + 1} p_i, i = 0, 1, ...$$

Equation (1) yields the following algorithm to sample from X :

Algorithm 1 (Poisson) *Input is a uniform random number u. The output is a random value from $X \sim Poisson(\lambda)$.*

1. *Set* $i = 0, p = e^{-\lambda}, F = p$
2. *If* $u < F$, *set* $X = i$ *and stop.*
3. *Update:* $p = \frac{\lambda}{i+1}p, F = F + p, i = i + 1$
4. *Go to step 2.*

At the ith step (iteration) of the algorithm, we find the ith subinterval of equation (1), and check if u belongs to that subinterval. The complexity of the algorithm depends on how many steps it needs to go through, i.e., how many subintervals it needs to compute, before stopping. Observe that if the algorithm outputs $X = i$, then it means the algorithm has gone through exactly $i + 1$, or, $X + 1$ steps. The average complexity of the algorithm, then, depends on the average value of $X + 1$. Therefore, we have

Lemma 1 *The expected number of steps in Algorithm (Poisson) is*

$$E[X + 1] = \lambda + 1.$$

Remark 1 *A more efficient algorithm would first compute the two integers that are closest to* λ, *say,* $k \leq \lambda < k+1$, *and then decide if* u *is less than, or greater than,* $p_0 + ... + p_k$. *If it is smaller, than start searching for the interval that contains* u *with* $[p_0 + ... + p_{k-1}, p_0 + ... + p_k)$, *and continue the search moving to the left. Otherwise, start with* $[p_0 + ... + p_k, p_0 + ... + p_{k+1})$ *and move to the right.*

1.5 Binomial distribution

First recall the definition of a Bernoulli trial; a random experiment whose outcome can be classified as either a "success" or a "failure", and the probability of "success" is a fixed value p. The probability density function of a Bernoulli random variable X is $P\{X = 1\} = p$ and $P\{X = 0\} = 1 - p$. Also recall that $E[X] = p$ and $Var(X) = p(1 - p)$. Now let's assume a Bernoulli trial is performed independently n times. For each trial, the probability of success is p. Let X be the number of successes obtained in these n trials. Clearly, the space of X is $0, 1, ..., n$. The density function of X is

$$P\{X = i\} = p_i = \binom{n}{i} p^i (1 - p)^{n-i}, i = 0, 1, ..., n. \tag{3}$$

We say X has the binomial distribution with parameters (n, p), if its density function is given by (3). Binomial random variables are usually denoted by $bin(n, p)$. Recall that if $X \sim bin(n, p)$, then

$$E[X] = np, Var(X) = np(1 - p).$$

We can compute p_i recursively using

$$p_{i+1} = P\{X = i + 1\} = \frac{n - i}{i + 1} \frac{p}{1 - p} P\{X = i\}.$$

The following algorithm makes use of this recursion in sampling a value from X :

Algorithm 2 (Binomial) *Input is a uniform random number u. The output is a random value from $X \sim bin(n,p)$.*

1. *Set $i = 0, c = \frac{p}{1-p}, p_i = (1-p)^n, F = p_i$*
2. *If $u < F$, set $X = i$ and stop.*
3. *Update: $p_i = \frac{n-i}{i+1} c p_i, F = F + p_i, i = i + 1$*
4. *Go to step 2.*

The average complexity of the algorithm is given by the following result.

Lemma 2 *The expected number of steps in Algorithm (Binomial) is*

$$E[X+1] = np + 1.$$

The following remark explains how the efficiency of the algorithm can be improved.

Remark 2
1. *If $p > 1/2$, then the probability of having a failure will be smaller. If we count the number of failures, instead of successes, then the algorithm will stop earlier (as we search from left to right) since having a large number of failures is less likely than having the same number of successes. Therefore, in this case, generate $Y = bin(n, 1-p)$ and set $X = n - Y$.*
2. *If $E[X] = np$ is large, then a better approach than searching from left to right will be to start around the mean of X, like we did in the case of Poisson distribution. Find the greatest integer less than or equal to np (i.e., $Floor(np)$), and compare this number with u. If $u < Floor(np)$, search to the left, otherwise search to the right.*

2 The Acceptance-Rejection Method

This method is due to von Neumann. It is used especially when the inverse transformation method is difficult to apply. Here is the idea in a nut shell: Let's assume we want to generate from the random variable X with density $\{p_i, i \geq 0\}$, and we have an efficient method to generate from another random variable Y with density $\{q_i, i \geq 0\}$. Is there a way to generate a value from Y and somehow use this value to generate a value from X? The answer is yes!

Idea Generate a value y from Y (density $\{q_i, i \geq 0\}$) and "accept" this as a value from X with probability p_y/q_y.

Intuitively, generating a value y from Y and accepting it as a value from X will occur with probability $q_y \times p_y/q_y$ which is p_y; the desired probability for the random variable X. The next algorithm and theorem will give a rigorous treatment of this idea.

We need a condition to be true for this idea to work: Let c be a constant such that $p_i/q_i \leq c$ for all i such that $p_i > 0$. In other words, we need an upper bound for the ratios of these probabilities.

Algorithm 3 (Acceptance-Rejection)

1. *Generate a value from Y with density $\{q_i, i \geq 0\}$. Call this value y.*
2. *Generate u from $U(0,1)$ (i.e., generate a uniform random number)*
3. *If $u < \dfrac{p_y}{cq_y}$, set $X = y$ and stop. Otherwise return to step 1.*

Example 1 *Let X be a random variable that takes values $1, 2, ..., 10$ with probabilities $0.11, 0.12, 0.09, 0.08, 0.12, 0.10, 0.09, 0.09, 0.10, 0.10$. Use the acceptance-rejection method to generate X.*

Solution 1 *First we need to find a suitable Y; a random variable with the same space as X, and a random variable we know how to generate. Let's try the discrete uniform distribution on {1,...,10} for Y. Density of Y is $q_i = 1/10$ for all i. We need to find an upper bound for p_i/q_i, where p_i are the probabilities given in the question, and q_i is 1/10. Simply choose $c = \max_{1 \leq i \leq 10} \frac{p_i}{1/10} = 1.2$. With these choices, the above algorithm simplifies as:*

1. *Generate u_1 from $U(0,1)$ and set $y = Floor(10u_1) + 1$ (this generates random variable Y; see Section 1.2)*
2. *Generate u_2 from $U(0,1)$*
3. *If $u_2 \leq \dfrac{p_y}{1.2\frac{1}{10}} = \frac{p_y}{0.12}$, set $X = y$. Otherwise return to step 1.*

It may be helpful to give an example on how the algorithm works: let's assume you obtain $u_1 = 0.21$ in step 1. Then $y = 3$. In step 2, let's assume we generate $u_2 = 0.74$. Compute $\frac{p_y}{0.12} = \frac{p_3}{0.12} = \frac{0.09}{0.12} = 0.75$. Since $u_2 = 0.74 < 0.75$, we set $X = 3$. In other words, 3 is the random value we get from the random variable X. If u_2 were larger than 0.75, then we would start all over again and go back to step 1.

We have not yet proved that the acceptance-rejection algorithm does what it claims. We do that next.

Theorem 2 *The acceptance-rejection algorithm generates a random variable X with density $\{p_i, i \geq 0\}$. In addition, the number of iterations of the algorithm needed to obtain a value from X is a geometric random variable with mean c.*

Proof. *First, let's compute the probability that in a single iteration Y takes the value i and this value is accepted.*

$$\begin{aligned} P\{Y &= i, \text{ it is accepted}\} = P\{Y=i\}P\{\text{accepted} \,|Y=i\} \qquad (4) \\ &= q_i \frac{p_i}{cq_i} = \frac{p_i}{c} \end{aligned}$$

Now let's compute the probability that a generated value in Step 1 is accepted in Step 3:

$$P\{\text{accepted}\} = \sum_i P\{\text{accepted}, Y=i\} = \sum_i \frac{p_i}{c} = \frac{1}{c},$$

where, in the second equality, we used equation (4). The above probability tells us something significant: each iteration independently gives an accepted value with probability $1/c$. Think about acceptance of a value as "success". Then, the number of iterations needed to get an accepted value is a geometric random variable with probability of success $1/c$. Then, the mean number of iterations to get an accepted value is c.

Finally, we prove that the density of X is $\{p_i, i \geq 0\}$.

$$\begin{aligned} P\{X &= i\} = \sum_{n=1}^{\infty} P\{i \text{ is accepted at the } n\text{th iteration}\} \\ &= \sum_{n=1}^{\infty} \left(1 - \frac{1}{c}\right)^{n-1} \frac{p_i}{c} \end{aligned}$$

where we use the fact that to be accepted at the nth iteration, all previous $(n-1)$ iterations need to be rejections, which occurs with probability $\left(1-\frac{1}{c}\right)^{n-1}$, and that the probability that at the nth iteration a value of i is generated and accepted is p_i/c, from equation (4). If we sum the geometric series, we get

$$P\{X=i\} = \frac{p_i}{c} \frac{1}{1-\left(1-\frac{1}{c}\right)} = p_i$$

which completes the proof. ■

Remark 3 *Note that since $p_i/q_i \leq c$ for all i, we have $p_i \leq cq_i$ and thus $\sum p_i = 1 \leq c \sum q_i = c$. So c is at least one. The average number of iterations needed in acceptance-rejection is c. Therefore the closer c is to 1, the more efficient is the acceptance-rejection method. Intuitively, if c is closer to 1, then that means the two densities $\{p_i\}$ and $\{q_i\}$ are "alike", or, $p_i \approx q_i$.*

Remark 4 *In Example 1, $c = 1.2$, so on the average the algorithm requires 1.2 iterations to obtain a value for X. This is almost as efficient as generating a discrete uniform random variable. Indeed, the reason for this efficiency is precisely because the density $\{p_i\}$ in this example was very "close" to the discrete uniform density $q_i = 1/10$.*

3 The Composition Method

Suppose we know how to simulate $\{p_i^1, i \geq 0\}$ and $\{p_i^2, i \geq 0\}$, and we want to simulate X with density

$$P\{X = i\} = \alpha p_i^1 + (1 - \alpha)p_i^2, 0 < \alpha < 1 \tag{5}$$

Observe the following fact: If X_1 and X_2 have densities $\{p_i^1, i \geq 0\}$ and $\{p_i^2, i \geq 0\}$, then the random variable defined by

$$X = \begin{cases} X_1 \text{ with probability } \alpha \\ X_2 \text{ with probability } 1 - \alpha \end{cases}$$

has the density given by (5). The following algorithm uses this fact.

Algorithm 4 (Composition) *1. Generate u from $U(0, 1)$*

2. If $u < \alpha$, generate a value from X_1. If $u > \alpha$, generate a value from X_2.

Example 2 *Simulate the random variable X with density*

$$p_i = P\{X = i\} = \begin{cases} 0.05, i = 1, 2, 3, 4, 5 \\ 0.15, i = 6, 7, 8, 9, 10 \end{cases}$$

We can represent X as the composition of two discrete uniform random variables which can be generated very efficiently. Indeed, let X_1 be the discrete uniform on $1, 2, ..., 10$, then $p_i^1 = 0.1$. And let X_2 be the random variable with the following density

$$p_i^2 = P\{X_2 = i\} = \begin{cases} 0, i = 1, 2, 3, 4, 5 \\ 0.2, i = 6, 7, 8, 9, 10 \end{cases}$$

Now define

$$X = \begin{cases} X_1 \text{ with probability } 0.5 \\ X_2 \text{ with probability } 0.5 \end{cases}$$

Note that the density function of X is

$$p_i = P\{X = i\} = P\{X_1 = i\}0.5 + P\{X_2 = i\}0.5$$

If $i = 1, 2, 3, 4, 5$, this simplifies to

$$p_i = P\{X_1 = i\}0.5 = 0.1 \times 0.5 = 0.05$$

since $P\{X_2 = i\} = 0$. And if $i = 6, 7, 8, 9, 10$, it simplifies to

$$p_i = P\{X_1 = i\}0.5 + P\{X_2 = i\}0.5 = 0.1 \times 0.5 + 0.2 \times 0.5 = 0.15$$

which is the desired density. For this example, the Algorithm (Composition) simplifies to

1. Generate u_1 and u_2 from $U(0, 1)$

2. *If* $u_1 < 0.5$, *set* $X = Floor(10u_2) + 1$. *Otherwise, set* $X = Floor(5u_2) + 6$. *(two independent uniform random numbers are used to assure the independence of two decisions: choosing which random variable to generate* (X_1 *or* X_2), *and generating from the chosen random variable.)*

Part II
Generating Continuous Distributions

4 The Inverse Transformation Method

Like in the discrete distribution case, the method is based on Theorem 1. To generate a random variable X with distribution function (c.d.f) F, we generate a uniform random number u from $U(0,1)$ and set $X = F^{-1}(u)$.

Next we will consider some special continuous distributions.

4.1 Exponential distribution

We say X is an exponential random variable with parameter $\lambda > 0$ and write $X \sim \exp(\lambda)$ if its density function is given by

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, \text{ if } x \geq 0 \\ 0, \text{ otherwise} \end{cases}.$$

The c.d.f. of X is

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, \text{ if } x \geq 0 \\ 0, \text{ otherwise} \end{cases}.$$

Recall that for $X \sim \exp(\lambda)$

$$E[X] = 1/\lambda \text{ and } Var(X) = 1/\lambda^2.$$

Note that the inverse of the c.d.f of $X \sim \exp(\lambda)$ is $F^{-1}(x) = -(1/\lambda)\log(1-x)$, $x > 0$. Also note that the distribution of $1-U$ and U are both $U(0,1)$, if U is $U(0,1)$. Combining these two observations, we conclude:

Algorithm 5 (Exponential) *Generating* $X \sim \exp(\lambda)$

1. *Generate* u *from* $U(0,1)$
2. *Set* $X = -\dfrac{\log u}{\lambda}$

4.2 Gamma distribution

We say X has a gamma distribution with parameters n and λ, and write $X \sim Gamma(n, \lambda)$, if its density function is

$$f(x) = \frac{\lambda^n}{\Gamma(n)} x^{n-1} e^{-\lambda x},\ 0 < x < \infty$$

where $\Gamma(n)$ is the gamma function defined by $\Gamma(n) = \int_0^\infty y^{n-1} e^{-y} dy,\ n > 0$. Recall that for $X \sim Gamma(n, \lambda)$,

$$E[X] = \alpha/\lambda,\ Var(X) = \alpha/\lambda^2.$$

Using the inverse transformation method to generate $X \sim Gamma(n, \lambda)$ requires finding the inverse of its c.d.f

$$F(x) = \int_0^x \frac{\lambda^n}{\Gamma(n)} y^{n-1} e^{-\lambda y} dy$$

This can be done using numerical methods. However, I will discuss a simpler method to generate X based on the following fact.

Fact $X \sim Gamma(n, \lambda)$ has the same distribution as the sum of n independent $X_n \sim \exp(\lambda)$ random variables.

Algorithm 6 (Gamma) *Generating* $X \sim Gamma(n, \lambda)$

1. *Generate independent random numbers* $u_1, ..., u_n$ *from* $U(0, 1)$
2. *Set* $X = -\frac{\log u_1}{\lambda} + ... + -\frac{\log u_n}{\lambda} = -\frac{1}{\lambda} \log(u_1 \cdot ... \cdot u_n)$

4.3 Normal distribution

We say X has normal distribution with parameters μ and σ^2, and write $X \sim \mathbf{N}(\mu, \sigma^2)$, if its density function is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty.$$

Here μ is any real constant, and σ is any positive real constant. The normal distribution is one of the most important distributions in probability and statistics, and it is widely used in financial mathematics.

To generate $X \sim \mathbf{N}(\mu, \sigma^2)$ using the inverse transformation method, we need to invert its distribution function

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(y-\mu)^2}{2\sigma^2}} dy.$$

This can be done fairly efficiently using numerical techniques. One such technique that is popular in the financial math literature is the Beasley-Springer-Moro algorithm (see pg 67-68 of Glasserman's book). You will write a code implementing this algorithm as homework. We will discuss other methods to generate the normal distribution later.

5 The Acceptance-Rejection Method

We want to generate a random variable X with density function $f(x)$. We assume that we know how to generate another random variable Y with density $g(x)$. Let c be a constant such that $f(x)/g(x) \leq c$ for all x in the space of X. The algorithm is similar to the discrete case:

Algorithm 7 (Acceptance-Rejection)

1. *Generate a value y from Y with density g*
2. *Generate u from $U(0,1)$*
3. *If $u \leq \frac{f(y)}{cg(y)}$, set $X = y$. Otherwise return to step 1.*

Theorem 3 *The random variable generated by Algorithm (Acceptance-Rejection) has density f. Moreover, the number of iterations of the algorithm that are needed is a geometric random variable with mean c.*
Proof. *The proof is similar to the discrete case.* ■

Example 3 *Let's use the acceptance-rejection method to generate X with density*

$$f(x) = 20x(1-x)^3, 0 < x < 1.$$

The space of X is $(0,1)$. Let's try a random variable with the same space that we use all the time, $U(0,1)$, as a candidate for Y. So, $g(x) = 1, 0 < x < 1$. We first need to find c; an upper bound for $f(x)/g(x) = 20x(1-x)^3$, when $0 < x < 1$. Using Calculus, we can show that the maximum value of $20x(1-x)^3$, when $0 < x < 1$, is $135/64 \approx 2.11$, which is attained when $x = 1/4$. Here is the algorithm:

1. *Generate u_1 and u_2 from $U(0,1)$*
2. *If $u_2 \leq \dfrac{20u_1(1-u_1)^3}{135/64} = \dfrac{256}{24}u_1(1-u_1)^3$, set $X = u_1$. Otherwise return to step 1.*

Note that the average number of steps (iterations) is $c = 135/64 \approx 2.11$.

Example 4 (Normal distribution) *Now we will discuss how to generate a normal random variable using the acceptance-rejection method. Let Z denote $\mathbf{N}(0,1)$, the standard normal variable. Consider $|Z|$, the absolute value of Z. Note that the density function of $|Z|$ is*

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, 0 < x < \infty.$$

Intuitively, since $|Z|$ takes a value x if $Z = x$ or $Z = -x$, the density of $|Z|$ at x is twice the density of Z. Also note that the space of $|Z|$ is, obviously, $(0, \infty)$.

We will use acceptance-rejection to generate $|Z|$, and then generate Z using the value we got from Z.

To generate $|Z|$, we need a candidate random variable Y. Let's try $Y \sim \exp(1)$, which has the same space as $|Z|$. The density function of Y is

$$g(x) = e^{-x}, x > 0.$$

Using Calculus, we find that the maximum of $f(x)/g(x) = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}+x}$ over $(0, \infty)$ occurs at the maximum of $x - x^2/2$ which is $\sqrt{2e/\pi} \approx 1.32$, and this maximum is attained when $x = 1$. The algorithm to generate $|Z|$ is:

1. *Generate a value y from $Y \sim \exp(1)$*
2. *Generate u from $U(0, 1)$*
3. *If $u \leq \frac{1}{\sqrt{2e/\pi}} \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}+x} = e^{x-x^2/2-1/2} = e^{-(x-1)^2/2}$, set $|Z| = y$. Otherwise, return to step 1.*

Finally, we generate Z from $|Z|$ as follows:

$$Z = \begin{cases} |Z| \text{ with probability 1/2} \\ -|Z| \text{ with probability 1/2} \end{cases}$$

6 Box-Muller Method

This is a popular method to generate a standard normal variables. Let X, Y be independent standard normal random variables and let R, θ be the polar coordinates of X, Y :

$$\begin{aligned} X &= R\cos\theta,\ Y = R\sin\theta \\ R^2 &= X^2 + Y^2, \theta = \tan^{-1} Y/X \end{aligned} \tag{6}$$

The joint density of X, Y is

$$f(x, y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} = \frac{1}{2\pi} e^{-\frac{(x^2+y^2)}{2}}.$$

We want to find the joint density of $d \equiv R^2$ and θ. The following fact helps us with that:

Fact Let $Y_1 = g_1(X_1, X_2)$ and $Y_2 = g_2(X_1, X_2)$, where X_i, Y_i are random variables. Then $f_{Y_1,Y_2}(y_1, y_2)$ (the joint density of Y_1, Y_2) and $f_{X_1,X_2}(x_1, x_2)$ (the joint density of X_1, X_2) satisfy

$$f_{Y_1,Y_2}(y_1, y_2) = f_{X_1,X_2}(x_1, x_2) \left|J(x_1, x_2)\right|^{-1},$$

where $J(x_1, x_2)$ is the Jacobian

$$J(x_1, x_2) = \begin{vmatrix} \partial g_1/\partial x_1 & \partial g_1/\partial x_2 \\ \partial g_2/\partial x_1 & \partial g_2/\partial x_2 \end{vmatrix}.$$

We compute the Jacobian for the Box-Muller transformation:

$$J = \begin{vmatrix} \partial d/\partial x & \partial d/\partial y \\ \partial\theta/\partial x & \partial\theta/\partial y \end{vmatrix} = \begin{vmatrix} 2x & 2y \\ \frac{-1/x^2}{1+y^2/x^2} & \frac{1/x}{1+y^2/x^2} \end{vmatrix} = 2$$

and we get

$$f(d,\theta) = \frac{1}{2\pi} e^{-\frac{(x^2+y^2)}{2}} \frac{1}{2} = \left(\frac{1}{2} e^{-d/2}\right)\left(\frac{1}{2\pi}\right)$$

Notice that $1/2\pi$ is the density function of $U(0, 2\pi)$ and $\frac{1}{2}e^{-d/2}$ is the density function for $\exp(1/2)$. Therefore this result shows that $d \equiv R^2$ and θ are independent (since their joint density is the product of marginal densities) and that $R^2 \sim \exp(1/2)$ & $\theta \sim U(0, 2\pi)$.

This gives us a method to generate standard normal random variables: first generate the polar coordinates, $R^2 \sim \exp(1/2)$ & $\theta \sim U(0, 2\pi)$, and then compute the independent standard normal random variables X, Y from Equation (6).

Algorithm 8 (Box-Muller) *Input: two independent random numbers from $U(0, 1)$. Output: two independent random numbers from the standard normal distribution.*

1. *Generate u_1, u_2 from $U(0, 1)$*
2. *Compute: $R^2 = -2\log u_1$ and $\theta = 2\pi u_2$*
3. $X = R\cos\theta = \sqrt{-2\log u_1}\cos(2\pi u_2)$
   $Y = R\sin\theta = \sqrt{-2\log u_1}\sin(2\pi u_2)$

Part III
Stochastic Processes

7 Brownian Motion

Definition 4 *The stochastic process $\{W(t); 0 \le t \le T\}$ is a standard Brownian motion on $[0, T]$ if*

1. $W(0) = 0$
2. *It has independent increments: for any $t_1, ..., t_n$, $W(t_2) - W(t_1)$, $W(t_3) - W(t_2)$, ..., $W(t_n) - W(t_{n-1})$ are independent random variables.*
3. *For every $0 \le s < t \le T$, $W(t) - W(s) \sim \mathbf{N}(0, t - s)$.*

Remark 5 *1. The path of a Brownian motion is continuous with probability 1, i.e., for a fixed w, $W(\cdot, w)$ is continuous. The path is nowhere differentiable and has infinite total variation with probability 1.*

2. *The distribution of* $W(t)-W(s)$ *depends only on the time increment* $t-s$. *So, for example,* $W(t) - W(s) \sim W(t-s)$.

Definition 5 *We say* $\{X(t); 0 \leq t \leq T\}$ *is a Brownian motion with drift* μ *and diffusion coefficient* σ^2 *if* $\frac{X(t)-\mu t}{\sigma}$ *is a standard Brownian motion. This process will be denoted by* $BM(\mu, \sigma^2)$. *The first two conditions of Definition 4 are still valid. The third condition is modified as: For every* $0 \leq s < t \leq T$, $X(t) - X(s) \sim \mathbf{N}(\mu(t-s), \sigma^2(t-s))$.

Historical remarks Robert Brown, a botanist, observed and described the movement of pollen grains immersed in liquid in 1827. Louis Bachelier introduced Brownian motion and used it to model stock prices (1900). Albert Einstein gave a mathematical analysis of Brownian motion in 1905. He showed that the movement can be explained by assuming that the pollen was continually being subjected to bombardment by the molecules of the surrounding medium. A rigorous mathematical analysis and its formal definition was given by Norber Wiener in 1918.

Bachelier's Model for Stock Prices Let $S(t)$ be the stock price at time t. The model assumes

$$S(t+y) - S(y) \sim \mathbf{N}(\mu t, \sigma^2 t) \tag{7}$$

and is independent of all prices up to time y.

Flaws of the model:

1. $S(t)$ can be negative
2. Consider two situations: (i) $S(0) = 10$ and we want $P\{S(1) - 10 = 10\}$, and (ii) $S(1) = 20$ and we want $P\{S(2) - 20 = 10\}$. In both probabilities the time increment is the same; 1 day. Therefore, the probabilities must be equal, if we accept the model (7). However, this does not sound right, intuitively: in the first case, we are computing the probability that the stock price has a 100% increase, and in the second case, we are computing the probability of a 50% increase. These events should not have the same probability.

Samuelson's Model This is the lognormal model we use today. It assumes

$$\log \frac{S(t+y)}{S(y)} \sim \mathbf{N}(\mu t, \sigma^2 t)$$

and is independent of all prices up to time y. This model corrects both flaws of the Bachelier model.

8 Simulating Brownian motion - the random walk approach

Given a fixed set of points in time, $0 < t_1 < ... < t_n$, we want to simulate values $W(t_1), ..., W(t_n)$. We will discuss two methods to simulate a Brownian motion; the random walk construction (also called the standard construction) which we discuss here, and the Brownian bridge construction we will discuss in the next section.

We will use the fact that

$$W(t_1) - W(t_0), W(t_2) - W(t_1), ..., W(t_n) - W(t_{n-1})$$

have distribution $\mathbf{N}(0, t_i - t_{i-1}), i = 1, ..., n$, and independent. Here $t_0 = 0$. Therefore, generate n independent standard normal random variables $Z_1, ..., Z_n$, using the methods we discussed in the previous chapter, and set:

$$\begin{aligned} W(t_1) &= \sqrt{t_1 - t_0} Z_1 \\ W(t_2) - W(t_1) &= \sqrt{t_2 - t_1} Z_2 \\ &\cdots \\ W(t_n) - W(t_{n-1}) &= \sqrt{t_n - t_{n-1}} Z_n \end{aligned} \tag{8}$$

If we want to simulate $BM(\mu, \sigma^2)$, then we modify equations (8) as

$$X(t_{i+1}) - X(t_i) = \mu(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i} Z_{i+1}$$

$i = 0, 1, ..., n - 1$.

Remark 6 *The Brownian motion $BM(\mu, \sigma^2)$ is the solution of the stochastic differential equation*

$$dX(t) = \mu dt + \sigma dW(t) \tag{9}$$

which means

$$\begin{aligned} X(t) &= X(0) + \int_0^t \mu ds + \sigma \int_0^t dW(s) \\ &\Rightarrow \quad X(t) = \mu t + \sigma W(t) \\ &\Rightarrow \quad \frac{X(t) - \mu t}{\sigma} = W(t) \end{aligned}$$

which is the definition of $X(t)$!

8.1 Brownian motion with time varying parameters $BM(\mu(t), \sigma^2(t))$

Generalize (9) as

$$dX(t) = \mu(t)dt + \sigma(t)dW(t) \text{ (Ito process)} \tag{10}$$

allowing drift and volatility terms vary deterministically by time. In its integral form, the above SDE is written as

$$X(t) = X(0) + \int_0^t \mu(s)ds + \int_0^t \sigma(s)dW(s).$$

The integral $I = \int_0^t \sigma(s)dW(s)$ is an Ito integral, with the following properties:

$$E[I] = 0, Var(I) = \int_0^t \sigma^2(s)ds$$

and I has the normal distribution. Then

$$X(t) - X(s) \sim \mathbf{N}\left(\int_s^t \mu(u)du, \int_s^t \sigma^2(u)du\right).$$

To simulate $BM(\mu(t), \sigma^2(t))$, we generate n independent standard normal random variables $Z_1, ..., Z_n$, and set

$$X(t_{i+1}) = X(t_i) + \int_{t_i}^{t_{i+1}} \mu(u)du + \sqrt{\int_{t_i}^{t_{i+1}} \sigma^2(u)du} Z_{i+1} \tag{11}$$

for $i = 0, 1, ..., n-1$.

Remark 7 *Consider the SDE (10)*

$$dX(t) = \mu(t)dt + \sigma(t)dW(t).$$

If we did not know its solution, then we would find an approximate solution by first discretizing the equation

$$X(t_{i+1}) - X(t_i) = \mu(t_i)(t_{i+1} - t_i) + \sigma(t_i)(W(t_{i+1}) - W(t_i))$$

and since $W(t_{i+1}) - W(t_i) \sim \sqrt{t_{i+1} - t_i} Z_{i+1}$,

$$X(t_{i+1}) = X(t_i) + \mu(t_i)(t_{i+1} - t_i) + \sigma(t_i)\sqrt{t_{i+1} - t_i} Z_{i+1} \tag{12}$$

for $i = 0, 1, ..., n-1$.

Compare equations (11) and (12). The first equation gives the values of the exact solution to the SDE (10) at t_i. *The second equation is obtained from a discretization of the SDE (10). Observe that these two equations do not even agree at* t_1 :

$$\begin{aligned} X(t_1) &= \int_0^{t_1} \mu(u)du + \sqrt{\int_0^{t_1} \sigma^2(u)du} Z_1 \text{ - exact solution} \\ X(t_1) &= \mu(0)t_1 + \sigma(0)\sqrt{t_1} Z_1 \text{ - discretized} \end{aligned}$$

The expectations and variances of $X(t_1)$ *are different.*

9 Other methods to simulate the standard Brownian motion

Let $Z_1, ..., Z_n$ be i.i.d. $\mathbf{N}(0,1)$ variables. Form the column vector $\mathbf{Z} = (Z_1, ..., Z_n)^T$, where T stands for the transpose operation. We say $\mathbf{Z}$ has the multivariate normal distribution with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{I}_n$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix. We use the notation $\mathbf{Z} \sim \mathbf{N}(\mathbf{0}, \mathbf{I}_n)$. Recall that, in general, $\mathbf{C} = (c_{ij})$ is the covariance matrix of a random vector $\mathbf{Y} = (Y_1, ..., Y_n)^T$ if $c_{ij} = Cov(Y_i, Y_j)$.

Now consider the vector $\mathbf{X} = \mathbf{AZ} + \boldsymbol{\mu}$ where $\mathbf{A}$ is an $n \times n$ matrix and $\boldsymbol{\mu}$ is an n dimensional vector. The ith component of the vector $\mathbf{X}$ is

$$X_i = a_{i1}Z_1 + ... + a_{in}Z_n + \mu_i.$$

Observe that

1. $X_i \sim \mathbf{N}(\mu_i, a_{i1}^2 + ... + a_{in}^2)$
2. $$\begin{aligned} Cov(X_i, X_j) &= Cov\left(\sum_k a_{ik}Z_k, \sum_{k'} a_{jk'}Z_{k'}\right) \\ &= \sum_k a_{ik} \sum_{k'} a_{jk'} Cov\left(Z_k, Z_{k'}\right) \end{aligned}$$

   and since $Cov\left(Z_k, Z_{k'}\right) = \delta_{kk'}$ (which is one if $k = k'$, and 0 otherwise), we get

   $$Cov(X_i, X_j) = \sum_k a_{ik}a_{jk} = (\mathbf{AA}^T)_{ij}.$$

   In other words, the covariance of X_i and X_j is the ijth entry of the matrix $\mathbf{AA}^T$.

We say $\mathbf{X}$ has multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{AA}^T$. In summary, if $\mathbf{Z} \sim \mathbf{N}(\mathbf{0}, \mathbf{I}_n)$ then $\mathbf{X} = \mathbf{AZ} + \boldsymbol{\mu} \sim \mathbf{N}(\boldsymbol{\mu}, \mathbf{AA}^T)$

Remark 8 *Recall that the mean vector and covariance matrix of a multivariate normal distribution uniquely determines the joint distribution of $X_1, ..., X_n$, through the moment generating function*

$$\begin{aligned} \phi(t_1, ..., t_n) &= E[\exp(t_1X_1 + ... + t_nX_n)] \\ &= \exp\left(\sum_i t_i\mu_i + \frac{1}{2}\sum_i \sum_j t_it_jCov(X_i, X_j)\right). \end{aligned}$$

Conclusion 6 *To simulate a multivariate normal vector $\mathbf{X} \sim \mathbf{N}(\boldsymbol{\mu}, \mathbf{C})$, do the following:*

1. *Generate $Z_1, ..., Z_n$ i.i.d. $\mathbf{N}(0,1)$ variables.*
2. *Compute $X_i = \sum_{j=1}^n a_{ij}Z_j + \mu_i, i = 1, ..., n$, or, in matrix notation, $\mathbf{X} = \mathbf{AZ} + \boldsymbol{\mu}$, where $\mathbf{AA}^T = \mathbf{C}$.*

9.1 Back to the random walk generation of $\{W(t), t \geq 0\}$

Given $t_1, ..., t_n$, we want to simulate $\mathbf{W} = (W(t_1), ..., W(t_n))$.

Lemma 3 $\mathbf{W}$ *has a multivariate normal distribution with* $E[W(t_i)] = 0$ *and* $Cov(W(t_i), W(t_j)) = \min(t_i, t_j)$.
Proof. *From the random walk construction discussed earlier, we have*

$$\begin{aligned} W(t_1) &= \sqrt{t_1} Z_1 \\ W(t_2) &= \sqrt{t_1} Z_1 + \sqrt{t_2 - t_1} Z_2 \\ &\cdots \\ W(t_n) &= \sqrt{t_1} Z_1 + ... + \sqrt{t_n - t_{n-1}} Z_n \end{aligned} \tag{13}$$

Then, $W(t_i)$ *is a linear combination of standard normals and thus* $\mathbf{W}$ *has a multivariate normal distribution. By the definition of* $W(t)$, $E[W(t_i)] = 0$ *for any* i. *Assume* $t_i < t_j$. *Then*

$$\begin{aligned} Cov(W(t_i), W(t_j)) &= Cov(W(t_i), W(t_j) - W(t_i) + W(t_i)) \\ &= Cov(W(t_i), W(t_j) - W(t_i)) + Cov(W(t_i), W(t_i)) \\ &= 0 + t_i \end{aligned}$$

The first covariance term is 0 from the independence of $W(t_i)$ *and* $W(t_j) - W(t_i)$. *The second covariance term is the variance of* $W(t_i)$ *which is* t_i. *This completes the proof.* ■

We have proved that $\mathbf{W} \sim \mathbf{N}(0, \mathbf{C})$ where $\mathbf{C} = (c_{ij})$ and $c_{ij} = \min(t_i, t_j)$. Now let $\mathbf{A}$ be the following lower triangular matrix:

$$\mathbf{A} = \begin{pmatrix} \sqrt{t_1} & 0 & 0 & \cdots & 0 \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & 0 & \cdots & 0 \\ \cdot & \cdot & & & \\ \cdot & \cdot & & & \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \sqrt{t_3 - t_2} & \cdots & \sqrt{t_n - t_{n-1}} \end{pmatrix}$$

Lemma 4 $\mathbf{A}\mathbf{A}^T = \mathbf{C}$ *and the random walk construction of* W *corresponds to computing* $\mathbf{W} = \mathbf{A}\mathbf{Z}$ *(see equation 13).*

Theorem 7 (Cholesky Factorization) *A symmetric positive definite matrix* D *can be uniquely written as* $D = GG^T$ *where* G *is a lower triangular matrix with positive diagonal entries. (A matrix* D *is positive definite if* $x^T Dx > 0$ *for all nonzero* x*)*

Let's use a simple example to show the factorization. Let $D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}$ and $G = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$. Multiply the matrices G and G^T, set the product to D,

and solve it for g_{ij} to get:

$$\begin{aligned} g_{11} &= \sqrt{d_{11}} \\ g_{21} &= d_{12}/\sqrt{d_{11}} = d_{12}/g_{11} \\ g_{22} &= \sqrt{d_{22} - \frac{d_{12}^2}{d_{11}}} = \sqrt{d_{22} - g_{21}^2} \end{aligned}$$

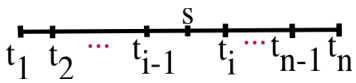
Remark 9 *Observe that* $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ *in the previous lemma is the Cholesky factorization of the covariance matrix* $\mathbf{C}$.

Remark 10 *See Glasserman for an algorithm to compute the Cholesky factorization of a matrix.*

9.2 The Brownian bridge construction of $W(t)$

The random walk construction generates $W(t_1), ..., W(t_n)$ in sequence; going from left to right. However, we can generate $W(t_i)$ in any order we want, if we know the conditional distributions. For example, consider the following order: $W(t_0) = 0, W(T), W(T/2), W(T/4), W(3T/4), ...$, filling in the midpoints of intervals recursively.

Let's assume for $0 < t_1 < ... < t_n$, we know the values for $W(t_1), ..., W(t_n)$. How can we generate $W(s)$, where $s \in (t_{i-1}, t_i)$?



To answer this, observe that given $W(t_{i-1})$, $W(s)$ is independent of all previous values $W(t_{i-2}), ..., W(t_1)$ from the independent increments property of Brownian motion. Similarly, given $W(t_i)$, $W(s)$ is independent of all values $W(t_{i+1}), ...W(t_n)$. In conclusion, to implement the Brownian bridge construction, or any construction where $W(t_i), i = 1, ..., n$ is generated not necessarily in the order of increasing time, we need to know the conditional distribution of $W(s)$ given the values $W(t_i)$ and $W(t_{i+1})$ where $t_i < t_{i+1}$.

Lemma 5 *The conditional distribution of* $W(s)$ *given that* $W(t_1) = x_1$ *and* $W(t_2) = x_2$ *where* $t_1 < s < t_2$ *is a normal distribution with*

$$\begin{aligned} E[W(s) \mid W(t_1) &= x_1 \& W(t_2) = x_2] = \frac{x_1(t_2 - s) + x_2(s - t_1)}{t_2 - t_1} \\ Var(W(s) \mid W(t_1) &= x_1 \& W(t_2) = x_2) = \frac{(t_2 - s)(s - t_1)}{t_2 - t_1} \end{aligned}$$

(note that the expectation is the linear interpolation ,in Lagrange form, for $(t_1, x_1), (t_2, x_2)$.

Proof. *Let* $f_{s|t_1,t_2}(x \mid x_1, x_2)$ *be the conditional density of* $W(s)$ *given that* $W(t_1) = x_1$ *and* $W(t_2) = x_2$. *Then*

$$f_{s|t_1,t_2}(x \mid x_1, x_2) = \frac{f_{s,t_1,t_2}(x, x_1, x_2)}{f_{t_1,t_2}(x_1, x_2)}$$

The event $W(s) = x, W(t_1) = x_1, W(t_2) = x_2$ *is equivalent to*

$$\begin{aligned} W(t_1) &= x_1 \\ W(s) - W(t_1) &= x - x_1 \\ W(t_2) - W(s) &= x_2 - x \end{aligned}$$

then, from independence of increments, the joint distributions can be written as

$$\frac{f_{s,t_1,t_2}(x, x_1, x_2)}{f_{t_1,t_2}(x_1, x_2)} = \frac{f_{t_1}(x_1) f_{s-t_1}(x - x_1) f_{t_2-s}(x_2 - x)}{f_{t_1}(x_1) f_{t_2-t_1}(x_2 - x_1)}$$

where f_t *denotes the density for* $\mathbf{N}(0, t)$. *The right-hand side of the above equation is*

$$\frac{\frac{1}{\sqrt{2\pi(s-t_1)}} \exp\left[-\frac{(x-x_1)^2}{2(s-t_1)}\right] \frac{1}{\sqrt{2\pi(t_2-s)}} \exp\left[-\frac{(x_2-x)^2}{2(t_2-s)}\right]}{\frac{1}{\sqrt{2\pi(t_2-t_1)}} \exp\left[-\frac{(x_2-x_1)^2}{2(t_2-t_1)}\right]}$$

$$= \underbrace{\frac{1}{\sqrt{2\pi}} \sqrt{\frac{t_2 - t_1}{(s-t_1)(t_2-s)}}}_{K} \exp\left[\underbrace{-\frac{1}{2}\left(\frac{(x-x_1)^2}{s-t_1} + \frac{(x_2-x)^2}{t_2-s} - \frac{(x_2-x_1)^2}{t_2-t_1} \right)}_{-\frac{1}{2}\left(x^2 - 2x \frac{x_1(t_2-s)+x_2(s-t_1)}{t_2-t_1}\right) \frac{t_2-t_1}{(t_2-s)(s-t_1)}} \right]$$

$$= K^* \exp\left[-\frac{1}{2} \frac{\left(x - \frac{x_1(t_2-s)+x_2(s-t_1)}{t_2-t_1}\right)^2}{\frac{(t_2-s)(s-t_1)}{t_2-t_1}} \right]$$

where in the last step we complete the square, and a new constant adds to K, *giving the final constant* K^*.

We know that the above function is a density function, so we can determine K^* *from the condition* $\int_{-\infty}^{\infty} f(x)dx = 1$. *Then, this density must be the density of a normal distribution with*

$$\begin{aligned} \mu &= \frac{x_1(t_2 - s) + x_2(s - t_1)}{t_2 - t_1} \\ \sigma^2 &= \frac{(t_2 - s)(s - t_1)}{t_2 - t_1} \end{aligned}$$

proving the lemma. ■

Now we are ready to describe the Brownian bridge construction. If we start with $S(0), S(T)$, and take the midpoint of $[0,T]$ for the next sample, and continue in this way, we obtain a simple construction. Let n be a power of 2. Then, repeated application of the lemma gives:

$$\begin{aligned} W(T) &= \sqrt{T}Z_1 \qquad (14)\\ W(T/2) &= \frac{1}{2}W(T) + \frac{\sqrt{T}}{2}Z_2 \\ W(T/4) &= \frac{1}{2}W(T/2) + \frac{\sqrt{2T}}{4}Z_3 \\ W(3T/4) &= \frac{1}{2}\left(W(T/2) + W(T)\right) + \frac{\sqrt{2T}}{4}Z_4 \\ &\cdots \\ W\left(\frac{(n-1)T}{n}\right) &= \frac{1}{2}\left[W\left(\frac{(n-2)T}{n}\right) + W(T)\right] + \sqrt{\frac{T}{2n}}Z_n \end{aligned}$$

For example, if $n = 2^3$, then Brownian bridge constructs $W(t_1), ..., W(t_8)$ in the following order:

$$W(t_8 = T), W(t_4 = T/2), W(t_2), W(t_6), W(t_1), W(t_3), W(t_5), W(t_7)$$

To verify equations (14), simply observe that when the points are chosen as midpoints, the conditional mean and variance of the lemma simplifies to

$$\begin{aligned} \mu &= \frac{x_1(t_2 - s) + x_2(s - t_1)}{t_2 - t_1} = \frac{x_1\frac{t_2-t_1}{2} + x_2\frac{t_2-t_1}{2}}{t_2 - t_1} = \frac{1}{2}(x_1 + x_2) \\ \sigma^2 &= \frac{(t_2 - s)(s - t_1)}{t_2 - t_1} = \frac{\frac{t_2-t_1}{2}\frac{t_2-t_1}{2}}{t_2 - t_1} = \frac{t_2 - t_1}{4} \end{aligned}$$

Remark 11 *See Glasserman, page 85, for an algorithm for the Brownian bridge construction.*

Remark 12 *The Brownian bridge construction does not offer any advantages in terms of error reduction over the random walk approach in Monte Carlo simulation. However, when it is used together with quasi-Monte Carlo (low-discrepancy) sequences, it may reduce error in some problems. See Caflisch et al. [1] and Papageorgiou [2].*

10 Geometric Brownian motion

$S(t)$ is a geometric Brownian motion if $\log S(t)$ is a BM with initial value $\log S(0)$, or, equivalently, if it can be written as $Ce^{X(t)}$ where $X(t)$ is a BM. Recall that $X(t) \sim BM(\mu, \sigma^2)$ satisfies the SDE

$$dX(t) = \mu dt + \sigma dW(t)$$

Now let's find the SDE satisfied by $S(t)$; the geometric Brownian motion given as $S(t) = S(0)e^{X(t)}$.

Ito's formula Let $F(X,t)$ be twice continuously differentiable function on $\mathbf{R}\times[0,T]$. Define a new stochastic process by $Y(t)=F(X(t),t)$. Then

$$dY(t)=\left[\frac{\partial F}{\partial t}+\mu\frac{\partial F}{\partial X}+\frac{1}{2}\sigma^2\frac{\partial^2 F}{\partial X^2}\right]dt+\left(\sigma\frac{\partial F}{\partial X}\right)dW(t) \qquad (15)$$

Put $F(X,t)=S(0)e^X$ in Ito's formula. We have

$$\frac{\partial F}{\partial t}=0,\frac{\partial F}{\partial X}=S(0)e^X,\frac{\partial^2 F}{\partial X^2}=S(0)e^X$$

then, from (15), the new process $S(t)=F(X(t),t)=S(0)e^{X(t)}$ satisfies

$$\begin{aligned} dS(t) &= \left[\mu S(0)e^X+\frac{1}{2}\sigma^2 S(0)e^X\right]dt+\left(\sigma S(0)e^X\right)dW(t) \\ &= S(0)e^X\left(\mu+\frac{\sigma^2}{2}\right)dt+S(0)e^X\sigma dW(t) \end{aligned}$$

and since $S(0)e^X=S(t)$, we can rewrite this as

$$\frac{dS(t)}{S(t)}=\left(\mu+\frac{\sigma^2}{2}\right)dt+\sigma dW(t).$$

However, in the literature, geometric BM's are often specified through the SDE

$$\frac{dS(t)}{S(t)}=\mu dt+\sigma dW(t).$$

Then, the underlying BM, $X(t)=\log S(t)$ will have drift $\mu-\frac{\sigma^2}{2}$, i.e.,

$$dX(t)=d\log S(t)=\left(\mu-\frac{\sigma^2}{2}\right)dt+\sigma dW(t).$$

You can prove this last equation by applying Ito's formula to $\log S(t)$.

In other words,

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

and

$$S(t)=S(0)e^{X(t)}=S(0)\exp\left(\left(\mu-\frac{\sigma^2}{2}\right)t+\sigma W(t)\right).$$

Definition 8 *We say $S(t)\sim GBM(\mu,\sigma^2)$ if S is given by*

$$\frac{dS(t)}{S(t)}=\mu dt+\sigma dW(t).$$

μ is called the drift parameter, although it is not the drift of $\log S(t)$ or $S(t)$. σ is called the volatility parameter.

10.1 Simulating GBM(μ, σ^2)

Given $0 = t_0 < t_1 < ... < t_n$, we generate values of $S(t)$ at $t_0, ..., t_n$ by

$$S(t_{i+1}) = S(t_i) \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i} Z_{i+1}\right) \tag{16}$$

where $Z_1, ..., Z_n$ are i.i.d. standard normals and $i = 0, ..., n-1$.

Compare the above recursion with the random walk generation of $BM(\mu, \sigma^2)$:

$$X(t_{i+1}) = X(t_i) + \mu(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i} Z_{i+1}$$

Taking the exp of both sides and replacing μ by $\mu - \frac{\sigma^2}{2}$ (as a result of applying Ito's formula), we obtain (16).

10.2 Properties of the lognormal distribution

If $S(t) \sim GBM(\mu, \sigma^2)$, then

$$\frac{S(t)}{S(0)} = \exp\left(\underbrace{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\sqrt{t}Z}_{\mathbf{N}((\mu - \sigma^2/2)t, \sigma^2 t)}\right) \tag{17}$$

Let's study these type of random variables in more detail.

Definition 9 *A random variable Y is said to have lognormal distribution with parameters μ, σ^2, and denoted by $Y \sim LN(\mu, \sigma^2)$, if $Y = e^X$ where $X \sim \mathbf{N}(\mu, \sigma^2)$.*

Let's compute the expectation and variance of Y. First recall the definition of the moment generating function $M(t)$ of a random variable X : $M_X(t) = E[e^{tX}]$. If $X \sim \mathbf{N}(\mu, \sigma^2)$, $M_X(t) = \exp(\mu t + \sigma^2 t^2/2)$. Then

$$E[Y] = E[e^X] = M_X(1) = e^{\mu + \sigma^2/2}$$

and

$$\begin{aligned} Var(Y) &= E[Y^2] - E[Y]^2 \\ &= E[e^{2X}] - \left(e^{\mu + \sigma^2/2}\right)^2 \\ &= M_X(2) - e^{2\mu + \sigma^2} \\ &= e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2} \\ &= e^{2\mu + \sigma^2}\left(e^{\sigma^2} - 1\right). \end{aligned}$$

The distribution function is

$$F_Y(y) = P\{Y \le y\} = P\{X \le \ln y\} = P\left\{Z \le \frac{\ln y - \mu}{\sigma}\right\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(\ln y - \mu)/\sigma} e^{-x^2/2} dx$$

and differentiation gives the density

$$f_Y(y) = \frac{1}{\sigma y\sqrt{2\pi}} \exp\left(-\frac{(\ln y - \mu)^2/\sigma^2}{2}\right).$$

Also note that the median of Y is e^{μ}, since $P\{Y \leq e^{\mu}\} = P\{X \leq \mu\} = 1/2$. Therefore the mean of Y is larger than its median, implying positive skewness.

From equation (17), we then conclude $S(t)/S(0) \sim LN\left((\mu - \sigma^2/2)t, \sigma^2 t\right)$. Therefore, from the properties we discussed above

$$\begin{aligned} E[S(t)/S(0)] &= \exp\left((\mu - \sigma^2/2)t + \sigma^2 t/2\right) = e^{\mu t} \\ &\Rightarrow \quad E[S(t)] = e^{\mu t} S(0) \end{aligned}$$

or, in general

$$E[S(t) \mid S(\tau), 0 \leq \tau \leq u] = E[S(t) \mid S(u)] = e^{\mu(t-u)} S(u)$$

Similarly,

$$Var(S(t)) = e^{2\mu t} S^2(0)\left(e^{\sigma^2 t} - 1\right).$$

Remark 13 *Although $E[S(t)]$ grows exponentially at rate μ, $S(t)$ grows at a lower rate. To this end, first observe that $\lim_{t\to\infty} \frac{W(t)}{t} = 0$ a.s. since $W(t)/t \sim \mathbf{N}(0, 1/t)$. Then*

$$\frac{1}{t}\log S(t) = \frac{\log S(0)}{t} + \frac{(\mu - \sigma^2/2)t + \sigma W(t)}{t} \to \mu - \sigma^2/2 \text{ a.s.}$$

1. *If $\mu - \sigma^2/2 > 0$, then $S(t) = \exp\left(\frac{\log S(t)}{t} t\right) \to \infty$ as $t \to \infty$, and if $\mu - \sigma^2/2 < 0$, then $S(t) \to 0$ as $t \to \infty$. If $\mu - \sigma^2/2 < 0 < \mu$, we get a "strange" behavior:*

$$\begin{aligned} S(t) &\to 0 \text{ a.s.} \\ E[S(t)] &= e^{\mu t} S(0) \to \infty \end{aligned}$$

   as $t \to \infty$. The explanation is skewness: rare but very large values of $S(t)$ make mean increase without bound.

2. $$\begin{aligned} &\frac{\log S(t)}{t} - (\mu - \sigma^2/2) \to 0 \text{ a.s.} \\ \Rightarrow\ &-\epsilon < \frac{\log S(t)}{t} - (\mu - \sigma^2/2) < \epsilon \text{ for } t > t^* \\ \Rightarrow\ &-\epsilon t < \log S(t) - (\mu - \sigma^2/2)t < \epsilon t \text{ for } t > t^* \\ \Rightarrow\ &\exp(-\epsilon t) < S(t)\exp(-(\mu - \sigma^2/2)t) < \exp(\epsilon t) \text{ for } t > t^* \\ \Rightarrow\ &\exp\left[\mu - (\sigma^2/2 + \epsilon)\right] t < S(t) < \exp\left[\mu - (\sigma^2/2 - \epsilon)\right] t < \exp[\mu t] \text{ for } t > t^* \end{aligned}$$

Example 5 *Pricing Asian arithmetic call options by Monte Carlo*

The payoff function is $(\bar{S} - K)^+$, *where* $\bar{S} = \frac{1}{m}\sum_{j=1}^{m} S(t_j)$, *and* K *is the exercise price. The option is observed at discrete times:* $0 = t_0 < t_1 < ... < t_m = T$. *The algorithm is:*

for $i = 1, ..., N$

for $j = 1, ..., m$

generate Z_{ij} *(*j*th independent draw from* $\mathbf{N}(0,1)$ *along the* i*th path)*

set $S_i(t_j) = S_i(t_{j-1}) \exp\left[(r - \sigma^2/2)(t_j - t_{j-1}) + \sigma\sqrt{t_j - t_{j-1}} Z_{ij}\right]$ *(this gives the* i*th price path* $S_i(t_1), ..., S_i(t_m)$*)*

set $\bar{S} = \frac{S_i(t_1)+...+S_i(t_m)}{m}$

set $C_i = e^{-rT}(\bar{S} - K)^+$

set $\hat{C}_N = \frac{C_1+...+C_N}{N}$

Note that here we use risk-neutral pricing and thus replace μ *by* r, *the risk-free interest rate. Also note that* N *is the number of price paths, and as* $N \to \infty$, $\hat{C}_N \to C$, *the true option price, almost surely.*

References

[1] Caflisch, R.E., Morokoff, W., Owen, A., 1997. Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. Journal of Computational Finance 1, 27-46.

[2] Papageorgiou, A., 2002. The Brownian Bridge Does Not Offer a Consistent Advantage in Quasi-Monte Carlo Integration, Journal of Complexity 18, 171-186.

11 Exercises

1. Write a computer program that implements the Inverse Transformation Method for a discrete density. The program should take the density function $\{p_1, p_2, ..., p_n\}$ as input, and output a random integer from $\{1, 2, ..., n\}$ with this density. To test your program apply it to the density $\{0.2, 0.2, 0.4, 0.1, 0.1\}$. Generate 1000 random integers in $\{1, 2, 3, 4, 5\}$ from this density, and plot a histogram for the data you obtain. Visually inspect whether the histogram is a good approximation of the theoretical density.

2. Let T be the number of arithmetical operations $(\pm, \times, \div)$ and comparisons (checking if a number is less than another) that need to be computed to generate a value from $X \sim Poisson(\lambda)$ using Algorithm (Poisson). Find $E[T]$ and $Var(T)$. (Ignore any operations in initialization and assignments $a := b$.)

3. A quantity X is computed by the following method:

   (a) Generate two independent uniform random numbers u, v.

   (b) If $u^2 + v^2 \geq 1$, return to step 1. Otherwise set $X = u$.

   Find the c.d.f. $F(x)$ for X. How many times will step 1 be performed? (Hint: Draw a quarter of a unit circle, inside a unit square. Pick an arbitrary x on the x-axis. Find the probability that $P\{X \leq x\}$ using geometric arguments via areas.)

4. Describe in detail how to simulate the value of a random variable X such that

$$P\{X = 1\} = .3, P\{X = 2\} = .2, P\{X = 3\} = .35, P\{X = 4\} = .15$$

   using the acceptance-rejection algorithm. What is the average complexity of the algorithm? In other words, find $E[T]$ where T is the number of arithmetical operations $(\pm, \times, \div)$ and comparisons (checking if a number is less than another) that need to be computed to simulate a value from X. (Ignore any operations in initialization and assignments $a := b$, and assume that generating a uniform random number is worth 3 arithmetical operations.

5. Write a computer program that implements the Box-Muller method. Your program will output independent standard normals X_1, X_2 when you input independent uniform variables U_1, U_2. Then generate 1000 pairs (X_{2i-1}, X_{2i}) $i = 1, ..., 1000$, and plot these pairs in $\mathbf{R}^2$, using the following two generators to compute the corresponding pairs (U_{2i-1}, U_{2i}) :

   (a) Mersenne twister,

   (b) a clearly poor LCG: $x_{n+1} \equiv 1229x_n + 1 (\mathrm{mod}\, 2048)$

   What conclusions do you make based on these graphs?

6. Write a computer program that implements the Box-Muller method, and the Beasley-Springer-Moro algorithm (see pg 67-68 of Glasserman's book). This algorithm gives an approximation to the inverse normal cdf. Then generate 2000 standard normals using this method. Test the normality of this data, and the Box-Muller data you generated in Problem 2 (a), using the Anderson-Darling test. (You will need a table for the percentiles for the Anderson-Darling statistic.) Which data set is closer to the normal distribution?

7. Give a method for generating the Weibull distribution function

$$F(x) = 1 - e^{-\alpha x^{\beta}}, 0 < x < \infty$$

8. Give a method for generating a random variable with density function

$$f(x) = \begin{cases} e^{2x}, -\infty < x < 0 \\ e^{-2x}, 0 < x < \infty \end{cases}$$

9. Suppose we know how to generate a random variable from any of the distributions $F_1, ..., F_n$. Describe how one can generate from the distribution $F(x) = \prod_{i=1}^{n} F_i(x)$.

10. Describe two ways of generating the distribution function $F(x) = x^n, 0 < x < 1$; one using the previous problem, another using the rejection algorithm (assuming that we know how to generate $F(x)$).

11. Verify the equation

$$W(3T/4) = \frac{1}{2}\left(W(T/2) + W(T)\right) + \frac{\sqrt{2T}}{4} Z_4$$

using Lemma 5.

12. Let $0 = t_0 < t_1 < ... < t_n = T$. Discuss how you would simulate $W(t)$ backwards in time, i.e., given $W(0) = 0$, simulate the Brownian motion in this order: $W(t_n), W(t_{n-1}), ..., W(t_1)$. You need to provide equations similar to (14).