1 Simulating normal (Gaussian) rvs with applications to simulating Brownian motion and geometric Brownian motion in one and two dimensions

Fundamental to many applications in financial engineering is the normal (Gaussian) distribution. It is the building block for simulating such basic stochastic processes as Brownian motion and geometric Brownian motion. In this section, we will go over algorithms for generating univariate normal rvs and learn how to use such algorithms for constructing sample paths of Brownian motion and geometric Brownian motion, in both one and two dimensions, at a desired sequence of times $t_1 < t_2 < \cdots < t_k$.

1.1 Generating a univariate normal random variable

We focus here on the generation of a rv $Z \sim N(0,1)$ for we can always then transform it into an $X \sim N(\mu, \sigma^2)$ via $X = \sigma Z + \mu$. For example, if $\{B(t) : t \geq 0\}$ denotes a standard Brownian motion (BM), then for any fixed $t > 0$, $B(t) \sim N(0,t)$ can be constructed via $B(t) = \sqrt{t}Z$. If it is desired to simulate the pair $(B(t_1), B(t_2))$ where $0 < t_1 < t_2$, then we can use the recursion $B(t_2) = B(t_1) + B(t_2) - B(t_1)$ and the basic stationary and independent increments properties of BM: Generate two iid $N(0,1)$ rvs, Z_1, Z_2 and set $B(t_1) = \sqrt{t_1}Z_1,\ B(t_2) = \sqrt{t_1}Z_1 + \sqrt{t_2 - t_1}Z_2$. The point is that $B(t_1) \sim N(0, t_1)$, and independently $B(t_2) - B(t_1) \sim N(0, t_2 - t_1)$. This method extends in the obvious fashion to the generation of the $k-$ dimensional multivariate normal vector $(B(t_1), B(t_2), \ldots, B(t_k))$, in which $0 < t_1 < t_2 < \cdots < t_k,\ k \geq 2$.

$N(0,1)$ density and cdf

The density of Z is given by

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}},\ x \in \mathcal{R},$$

and the cdf by

$$\Theta(x) = P(Z \leq x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-y^2}{2}} dy,\ x \in \mathcal{R}.$$

Using the inverse transform method is not possible here since we do not have an explicit closed formula for $\Theta^{-1}(y)$. One could of course approximate this inverse function by some closed form function and then use that, and there are some algorithms in the literature for doing that. But it is possible to exactly generate a $Z \sim N(0,1)$ using alternative (clever) algorithms, so that is what we will do.

Acceptance rejection algorithm for Z

Recall from our lecture on the acceptance rejection method that we already have an algorithm for generating a rv $Z \sim N(0,1)$:

Algorithm for generating $Z \sim N(0,1)$ via the acceptance rejection method

1. Generate two independent exponentials at rate 1; $Y_1 = -\ln(U_1)$ and $Y_2 = -\ln(U_2)$.

2. If $Y_2 \geq (Y_1 - 1)^2/2$, set $|Z| = Y_1$; otherwise go back to 1.

3. Generate U. Set $Z = |Z|$ if $U \leq 0.5$, set $Z = -|Z|$ if $U > 0.5$.

On average, this algorithm requires $2\sqrt{2e/\pi} + 1 \approx 3.64$ uniforms to produce one copy of Z. Since we of course will use the algorithm many many times over and over again, we can do even better by using the "free" independent exponential produced as overshoot in a success at step *2)*, $Y \overset{\text{def}}{=} Y_2 - (Y_1 - 1)^2/2$, as the Y_1 in step *1)* in the next round; this reduces the expected number to 2.64.

We next introduce another algorithm that generates pairs Z_1, Z_2 of independent normals each time. Called the *polar method*, it requires some more complex calculations (sin, cos, etc.) but has an advantage of only requiring one uniform for each copy of Z, not a random number. In a variety of real applications, it is very desirable to try to "synchronize" the use of random numbers across different runs in a simulation so as to reduce the variance of an estimate, and thus a non-random algorithm might well be useful at times. So we include it here in our study.

Algorithm for generating $Z \sim N(0,1)$ via the polar method

1. Generate an exponential at rate 1/2, $D = -2\ln(U_1)$, and a uniform over $(0, 2\pi)$, $\Theta = 2\pi U_2$.

2. Set $Z_1 = \sqrt{D}\cos\Theta,\ Z_2 = \sqrt{D}\sin\Theta$.

The polar method is easily derived (proved) via starting with an iid $N(0,1)$ pair Z_1, Z_2, mapping the vector (Z_1, Z_2) into polar coordinates, $D = Z_1^2 + Z_2^2$ (the squared distance from the origin), $\Theta = \arctan(Z_2/Z_1)$ (the angle, counterclockwise, from the horizontal axis to the point (Z_1, Z_2)), then deriving (using elementary multivariate calculus, with the Jacobian, etc.) the joint density function $g(d, \theta)$ of (D, Θ), to reveal that D is an exponential at rate 1/2 and independent of D, the angle Θ is uniform over $(0, 2\pi)$; $g(d, \theta) = \frac{1}{2}e^{-d/2} \times \frac{1}{2\pi},\ d > 0,\ \theta \in (0, 2\pi)$. Using this fact inversely then yields the algorithm: we can produce Z_1, Z_2 by mapping an independent pair (D, Θ) back into rectangular coordinates; that is exactly what step *2.* in the algorithm does.

1.2 Simulating 1-dimensional Brownian motion (BM) and geometric Brownian motion (GBM)

1.2.1 Standard BM

A stochastic process $\mathbf{B} = \{B(t) : t \geq 0\}$ possessing (wp1) continuous sample paths is called standard Brownian motion (BM) if

1. $B(0) = 0$.

2. $\mathbf{B}$ has both stationary and independent increments.

3. $B(t) - B(s)$ has a normal distribution with mean 0 and variance $t - s,\ 0 \leq s < t$.

2) and *3)* together can be summarized by: If $t_0 = 0 < t_1 < t_2 < \cdots < t_k$, then the increment rvs $B(t_i) - B(t_{i-1}),\ i \in \{1, \ldots k\}$, are independent and distributed as $N(0, t_i - t_{i-1})$. In particular, $B(t_i) - B(t_{i-1})$ is independent of $B(t_{i-1}) = B(t_{i-1}) - B(0)$.

If we only wish to simulate $B(t)$ at one fixed value $t > 0$, then we need only generate a unit normal $Z \sim N(0,1)$ and set $B(t) = \sqrt{t}Z$. But typically, we will want to simulate (say) k values at times $t_0 = 0 < t_1 < t_2 < \cdots < t_k$ and we can easily do so as follows:

Generate k iid unit normals $Z_1, Z_2, \dots, Z_k$, then construct the independent increments via $B(t_i) - B(t_{i-1}) = \sqrt{t_i - t_{i-1}} Z_i,\ i = 1, \dots, k$.

Thus to simulate the values $B(t_1), \dots, B(t_k)$, we sequentially generate unit normals, $Z_1, Z_2, \dots, Z_k$, and use the recursion
$B(t_{i+1}) = B(t_i) + (B(t_{i+1}) - B(t_i)) = B(t_i) + \sqrt{t_{i+1} - t_i} Z_{i+1},\ i \in \{0, \dots k-1\}$.

Simulating Standard BM at times $0 = t_0 < t_1 < t_2 < \cdots < t_k$*:*

Sequentially generate unit normals $Z_1, Z_2, \dots, Z_k$, and recursively define

$$\begin{aligned} B(t_1) &= \sqrt{t_1} Z_1 \\ B(t_2) &= B(t_1) + \sqrt{t_2 - t_1} Z_2 = \sqrt{t_1} Z_1 + \sqrt{t_2 - t_1} Z_2 \\ &\vdots \\ B(t_k) &= \sum_{i=1}^{k} \sqrt{t_i - t_{i-1}} Z_i. \end{aligned}$$

In the end, we see that to simulate BM at a collection of specific times we need only generate unit normals.

1.2.2 BM with drift

$X(t) = \sigma B(t) + \mu t$ will denote the BM with drift μ and variance term $\sigma > 0$. It has continuous sample paths and is defined by

1. $X(0) = 0$.

2. $\mathbf{X}$ has both stationary and independent increments.

3. $X(t) - X(s)$ has a normal distribution with mean $\mu(t-s)$ and variance $\sigma^2(t-s),\ 0 \le s < t$.

$X(t) - X(s)$ thus can be constructed (simulated) by generating a standard normal rv Z and setting $X(t) - X(s) = \sigma\sqrt{t-s} Z + \mu(t-s)$. Again, by the stationary and independent increments, we can simulate such a BM at times $0 = t_0 < t_1 < t_2 < \cdots < t_k$, by generating k iid unit normals $Z_1, Z_2, \dots, Z_k$ and using the recursion
$X(t_{i+1}) = X(t_i) + (X(t_{i+1}) - X(t_i)) = X(t_i) + \sigma\sqrt{t_{i+1} - t_i} Z_{i+1} + \mu(t_{i+1} - t_i)$.

Simulating BM with drift μ *and variance term* σ *at times* $0 = t_0 < t_1 < t_2 < \cdots < t_k$*:*

Sequentially generate unit normals $Z_1, Z_2, \dots, Z_k$, and recursively define

$$\begin{aligned} X(t_1) &= \sigma\sqrt{t_1} Z_1 + \mu t_1 \\ X(t_2) &= X(t_1) + \sigma\sqrt{t_2 - t_1} Z_2 + \mu(t_2 - t_1) = \sigma\sqrt{t_1} Z_1 + \mu t_1 + \sigma\sqrt{t_2 - t_1} Z_2 + \mu(t_2 - t_1) \\ &\vdots \\ X(t_k) &= \sum_{i=1}^{k} (\sigma\sqrt{t_i - t_{i-1}} Z_i + \mu(t_i - t_{i-1})). \end{aligned}$$

1.2.3 Geometric BM

Geometric Brownian motion (GBM) is given by

$$S(t) = S(0)e^{X(t)},\ t \geq 0,$$

where $X(t) = \sigma B(t) + \mu t,\ t \geq 0$, is a BM. $e^{X(t)}$ has a lognormal distribution for each fixed $t > 0$. In general if $Y = e^X$ is lognormal with $X \sim N(\mu, \sigma^2)$, then we can easily simulate Y via setting $Y = e^{\sigma Z + \mu}$, with $Z \sim N(0,1)$.

Moreover, for any $0 \leq s < t$ it holds that

$$S(t) = S(0)\frac{S(s)}{S(0)} \times \frac{S(t)}{S(s)} = S(0)e^{X(s)} \times e^{X(t)-X(s)},$$

and since the increment $X(s)$ is independent of the increment $X(t) - X(s)$, we conclude that the consecutive ratios $\frac{S(s)}{S(0)}$ and $\frac{S(t)}{S(s)}$ are independent lognormals. We can thus simulate the pair $(S(s), S(t))$ by generating two iid $N(0,1)$ rvs, Z_1, Z_2 and setting $S(s) = S(0)e^{\sigma\sqrt{s}Z_1+\mu s}$, $S(t) = S(s)e^{\sigma\sqrt{t-s}Z_2+\mu(t-s)} = S(0)e^{\sigma\sqrt{s}Z_1+\mu s} \times e^{\sigma\sqrt{t-s}Z_2+\mu(t-s)}$.

More generally, for $0 = t_0 < t_1 < t_2 < \cdots < t_k$, define $Y_i = S(t_i)/S(t_{i-1}),\ i \in \{1, 2, \ldots, k\}$. Then we can write

$$\begin{aligned} S(t_1) &= S(0)Y_1 \\ S(t_2) &= S(t_1)Y_2 = S_0 Y_1 \times Y_2 \\ &\vdots \\ S(t_k) &= S(t_{k-1})Y_k = S_0 Y_1 \times Y_2 \times \cdots \times Y_k. \end{aligned}$$

The Y_i are independent lognormal rvs and can be constructed by generating k iid $N(0,1)$ rvs, $Z_1, Z_2, \ldots Z_k$ and setting

$$Y_i = e^{\sigma\sqrt{t_i - t_{i-1}}Z_i + \mu(t_i - t_{i-1})},\ i \in \{1, 2, \ldots, k\}. \tag{1}$$

Simulating paths of GBM is thus an easy consequence of our algorithm for simulating paths of BM since for $0 = t_0 < t_1 < t_2 < \cdots < t_k$, the following recursion holds

$$S(t_{i+1}) = S(t_i)e^{X(t_{i+1})-X(t_i)},\ i \in \{0, 1, \ldots, k-1\}.$$

Simulating Geometric BM (with drift μ and variance term σ) at times $0 = t_0 < t_1 < t_2 < \cdots < t_k$:

Sequentially generate unit normals $Z_1, Z_2, \ldots, Z_k$, and set the Y_i as in (1). Then recursively define

$$\begin{aligned} S(t_1) &= S(0)Y_1 \\ S(t_2) &= S(t_1)Y_2 = S_0 Y_1 \times Y_2 \\ &\vdots \\ S(t_k) &= S(t_{k-1})Y_k = S_0 Y_1 \times Y_2 \times \cdots \times Y_k. \end{aligned}$$

1.3 Simulating correlated BM and GBM in two dimensions in which the two BM's have a specific desired correlation ρ.

In many financial applications, there are several correlated assets that make up a portfolio or from which certain options/derivatives are created (spread options, basket options, etc.). Here we focus on the case of two correlated assets.

Let $W_1(t)$ and $W_1(t)$ denote standard Brownian motions. Consider two Brownian motions $X_1(t) = \sigma_1 W_1(t) + \mu_1 t$, and $X_2(t) = \sigma_2 W_1(t) + \mu_2 t$. $\mathbf{X}(t) = (X_1(t), X_2(t))^{\mathbf{T}}$ is a two-dimensional Brownian motion, where we shall assume the coordinates have correlation coefficient ρ: For a given $-1 < \rho < 1$,

$$\frac{Cov(X_1(t), X_2(t))}{\sigma_1\sqrt{t} \times \sigma_2\sqrt{t}} = \rho, \ t > 0. \tag{2}$$

To construct this BM, we start with two independent standard BM's, $B_1(t)$ and $B_2(t)$, define $\mathbf{B}(t) = (B_1(t), B_2(t))^{\mathbf{T}}$, define the 2×2 matrix

$$\mathbf{A} = \left(\begin{array}{cc} \sigma_1 & 0 \\ \sigma_2\rho & \sigma_2\sqrt{1-\rho^2} \end{array} \right),$$

and construct

$$\mathbf{X}(t) = \mathbf{AB(t)} + \mu t, \ t \geq 0,$$

where $\mu = (\mu_1, \mu_2)^{\mathbf{T}}$.

Letting

$$\mathbf{\Sigma} = \mathbf{AA^T} = \left(\begin{array}{cc} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{array} \right),$$

we say that $\mathbf{X}$ is a two-dimensional BM with drift vector μ and covariance matrix $\mathbf{\Sigma}$; we denote this by

$$\mathbf{X} \sim BM(\mu, \mathbf{\Sigma}).$$

From (2), this implies that for each $t > 0$, $Cov(X_1(t), X_2(t)) = \rho\sigma_1\sigma_2 t$.

As with BM in 1-dimension, $\mathbf{X}$ has both stationary and independent increments.

If we wish to simulate $\mathbf{X}(t)$ at a fixed time $t > 0$, then we can generate Z_1, Z_2 iid $N(0,1)$, define the vector $\mathbf{Z} = (Z_1, Z_2)^{\mathbf{T}}$, then set

$$\mathbf{X}(t) = \sqrt{t}\mathbf{AZ} + \mu t.$$

For generating the pair $\mathbf{X}(t_1)$ and $\mathbf{X}(t_2)$ at the times $0 < t_1 < t_2$, we generate a pair of $N(0,1)$ vectors, $\mathbf{Z}_1 = (Z_{1,1}, Z_{1,2})^{\mathbf{T}}$, $\mathbf{Z}_2 = (Z_{2,1}, Z_{2,2})^{\mathbf{T}}$ and define

$$\begin{aligned} \mathbf{X}(t_1) &= \sqrt{t_1}\mathbf{AZ_1} + \mu t_1. \\ \mathbf{X}(t_2) &= \mathbf{X}(t_1) + \sqrt{t_2 - t_1}\mathbf{AZ_2} + \mu(t_2 - t_1) = \sqrt{t_1}\mathbf{AZ_1} + \mu t_1 + \sqrt{t_2 - t_1}\mathbf{AZ_2} + \mu(t_2 - t_1). \end{aligned}$$

This then yields

An algorithm for simulating $\mathbf{X} \sim BM(\mu, \mathbf{\Sigma})$ at times $0 = t_0 < t_1 < t_2 < \cdots < t_k$:

Sequentially generate k iid pairs of independent $N(0,1)$ rvs; $\mathbf{Z}_1 = (Z_{1,1}, Z_{1,2})^{\mathbf{T}}$, $\mathbf{Z}_2 = (Z_{2,1}, Z_{2,2})^{\mathbf{T}}$, $\ldots, \mathbf{Z}_k = (Z_{k,1}, Z_{k,2})^{\mathbf{T}}$, then recursively define

$$\begin{aligned}
\mathbf{X}(t_1) &= \sqrt{t_1}\mathbf{A}\mathbf{Z_1} + \mu t_1. \\
\mathbf{X}(t_2) &= \mathbf{X}(t_1) + \sqrt{t_2 - t_1}\mathbf{A}\mathbf{Z_2} + \mu(t_2 - t_1) = \sqrt{t_1}\mathbf{A}\mathbf{Z_1} + \mu t_1 + \sqrt{t_2 - t_1}\mathbf{A}\mathbf{Z_2} + \mu(t_2 - t_1). \\
&\vdots \\
\mathbf{X}(t_k) &= \mathbf{X}(t_{k-1}) + \sqrt{t_k - t_{k-1}}\mathbf{A}\mathbf{Z_k} + \mu(t_k - t_{k-1}) = \sum_{i=1}^{k}(\sqrt{t_i - t_{i-1}}\mathbf{A}\mathbf{Z_i} + \mu(t_i - t_{i-1})).
\end{aligned}$$

From here we can then define correlated geometric BM's (GBM)

$$S_1(t) = S_1(0)e^{X_1(t)}, \ S_2(t) = S_2(0)e^{X_2(t)}.$$

To simulate these, we use the algorithm above for the BM, coordinate by coordinate, utilizing the rows of the matrix $\mathbf{A} = (A_{i,j})$ given above, together with the algorithm at the end of Section 1.2.3 for simulating a one dimensional GBM.

An algorithm for simulating two correlated GBMs with underlying two-dimensional $\mathbf{X} \sim BM(\mu, \mathbf{\Sigma})$ at times $0 = t_0 < t_1 < t_2 < \cdots < t_k$:

Sequentially generate k iid pairs of independent $N(0,1)$ rvs; $\mathbf{Z}_1 = (Z_{1,1}, Z_{1,2})^{\mathbf{T}}$, $\mathbf{Z}_2 = (Z_{2,1}, Z_{2,2})^{\mathbf{T}}$, $\ldots, \mathbf{Z}_k = (Z_{k,1}, Z_{k,2})^{\mathbf{T}}$, then recursively define

$$\begin{aligned}
S_1(t_1) &= S_1(0)e^{\sqrt{t_1}\left(\sum_{j=1}^{2} A_{1,j}Z_{1,j}\right) + \mu_1 t_1}. \\
S_1(t_2) &= S_1(t_1)e^{\sqrt{t_2 - t_1}\left(\sum_{j=1}^{2} A_{1,j}Z_{2,j}\right) + \mu_1(t_2 - t_1)}. \\
&\vdots \\
S_1(t_k) &= S_1(t_{k-1})e^{\sqrt{t_k - t_{k-1}}\left(\sum_{j=1}^{2} A_{1,j}Z_{k,j}\right) + \mu_1(t_k - t_{k-1})}.
\end{aligned}$$

$$\begin{aligned}
S_2(t_1) &= S_2(0)e^{\sqrt{t_1}\left(\sum_{j=1}^{2} A_{2,j}Z_{1,j}\right) + \mu_2 t_1}. \\
S_2(t_2) &= S_2(t_1)e^{\sqrt{t_2 - t_1}\left(\sum_{j=1}^{2} A_{2,j}Z_{2,j}\right) + \mu_2(t_2 - t_1)}. \\
&\vdots \\
S_2(t_k) &= S_2(t_{k-1})e^{\sqrt{t_k - t_{k-1}}\left(\sum_{j=1}^{2} A_{2,j}Z_{k,j}\right) + \mu_2(t_k - t_{k-1})}.
\end{aligned}$$

Remark 1.1 The case of three or more dimensions can be handled analogously to our method here; we will cover this in more detail later.