Simulating Stochastic Differential Equations

1 Brief Review of Stochastic Calculus and Itô's Lemma

Let S_t be the time t price of a particular stock. We know that if $S_t \sim GBM(\mu, \sigma^2)$, then

$$S_t = S_0 \; e^{(\mu - \sigma^2/2)t + \sigma B_t} \tag{1}$$

where B_t is the Brownian motion driving the stock price. An alternative possibility is to use a *stochastic differential equation* (SDE) to describe the evolution of S_t. In this case we would write

$$S_t = S_0 + \int_0^t \mu S_u \; du + \int_0^t \sigma S_u \; dB_u \tag{2}$$

or in short-hand,

$$dS_t = \mu S_t \; dt + \sigma S_t \; dB_t. \tag{3}$$

A number of observations are in order:

(1) The SDE defined by (2) can be shown to be well-defined. In particular, while the first integral on the right-hand-side of (2) is a regular Riemann integral, the second integral is a *stochastic integral*. Without going into any technical details, it is convenient to interpret this integral as

$$\int_0^t \sigma S_u \; dB_u = \lim_{h \to 0} \sum \sigma S_{t_{i-1}} (B_{t_i} - B_{t_{i-1}}) \tag{4}$$

where $h = \max_i |t_i - t_{i-1}|$ is the width of the *partition*. The important feature of (4) is that the S_t terms are evaluated at the *left-hand* point of the intervals. This feature is extremely important in finance as it may be interpreted as modelling the inability of people to see into the future.

In general, we can similarly interpret the stochastic integral, $\int X(u, B_u) \; dB_u$, so that

$$\int_0^t X(u, B_u) \; dB_u = \lim_{h \to 0} \sum X(t_{i-1}, B_{t_{i-1}})(B_{t_i} - B_{t_{i-1}}).$$

(2) At this point, it is not clear that (1) and (2) define the same process but we will soon see that this is indeed the case.

(3) It is important to note that on its own, equation (3) has no meaning. It is only shorthand for equation (2).

In general, it is often convenient to model stock prices and interest rates as SDE's. Another example is given by the assumption that $X_t = \log(S_t)$ is an *Ornstein-Uhlenbeck* (OU) process. In particular, this means that

$$dX_t = -\gamma(X_t - \alpha) \; dt + \sigma dB_t \tag{5}$$

where γ, α and σ are non-negative constants.

Exercise 1 *In what circumstances might an Ornstein-Uhlenbeck model prove useful?*

Solving Stochastic Differential Equations

As is the case with ODE's and PDE's, we would like to be able to solve SDE's. The most useful tool for doing this is Itô's Lemma which we now state.

Itô's Lemma: Suppose x_t is an Itô process with $dx_t = a(x,t)\ dt + b(x,t)\ dB_t$. Let $y_t = F(x,t)$. Then

$$dy_t = \left(\frac{\partial F}{\partial x}\ a\ +\ \frac{\partial F}{\partial t}\ +\ \frac{1}{2}\ \frac{\partial^2 F}{\partial x^2}\ b^2 \right)\ dt\ +\ \frac{\partial F}{\partial x}\ b\ dB_t. \tag{6}$$

Exercise 2 *Use Itôs Lemma to see that (1) is indeed a solution to (2).*

Exercise 3 *Apply Itôs Lemma to* $\exp(\gamma t)X_t$ *in the OU model to see that*

$$S_T = \exp\left(\alpha + \exp(-\gamma T)[X_0 - \alpha] + \sigma \exp(-\gamma T) \int_0^T \exp(-\gamma s)\ dB_s \right). \tag{7}$$

---

2 Simulating Stochastic Differential Equations: Diffusions

The following three examples provide motivation for why we often need to simulate stochastic differential equations in order to estimate quantities of interest.

Example 1 (Geometric Brownian Motion)

We want to compute $\theta := \mathrm{E}[f(X_T)]$ where X_t satisfies

$$dX_t\ =\ \mu X_t\ dt + \sigma X_t\ dB_t. \tag{8}$$

The solution to (8) is of course given by

$$X_T\ =\ X_0 \exp\left((\mu - \sigma^2/2)T + \sigma B_T \right). \tag{9}$$

We recognize that X_T depends on the Brownian motion only through the Brownian motion's terminal value, B_T. This implies that even if we are unable to compute θ analytically, we can estimate it by simulating B_T directly. (In this case it is also true that since the distribution of X_T is known, we could also simulate X_T directly. And of course, as an alternative to simulation, we could choose to estimate θ by evaluating the expectation numerically.) ∎

Example 2 (OU Process)

Suppose now that we want to compute $\theta := \mathrm{E}[f(X_T)]$ where X_t satisfies

$$dX_t = -\gamma(X_t - \alpha)\ dt + \sigma\ dB_t. \tag{10}$$

The solution to (10) is given by

$$X_T\ =\ \alpha + \exp(-\gamma T)[X_0 - \alpha] + \sigma \exp(-\gamma T) \int_0^T \exp(-\gamma s)\ dB_s \tag{11}$$

Note that unlike the previous example, X_T now depends on the entire path of the Brownian motion. This means that we cannot compute an unbiased estimate of θ by first simulating the entire path of the Brownian motion since it is only possible to simulate the latter at discrete intervals of time. It so happens, however, that we know the distribution of X_T: it is normal. In particular, this places us back in the context of Example 1 where, if θ cannot be computed analytically, we can estimate it by either simulating X_T directly or by evaluating the expectation numerically. ■

Example 3 (CIR Model with Time Varying Parameters)

Again we want to compute $\theta := \mathrm{E}[f(X_T)]$ where X_t satisfies

$$dX_t \;=\; \alpha(\mu(t) - X_t)\;dt \;+\; \sigma\sqrt{X_t}\;dB_t. \tag{12}$$

and where $\mu(t)$ is a deterministic function of time. While we recognize that X_t follows a CIR process with time varying parameters, we do not know how to find an explicit solution[1] to the SDE in (12).

Of course we do not necessarily need an explicit solution to (12) to determine the distribution of X_T (which is what we need to evaluate θ). For example, in the CIR model with constant parameters, we still do not have an explicit solution to the SDE yet it is known that X_T has a non-central χ^2 distribution from which we can easily simulate. Unfortunately, however, once we move to a CIR model with time-varying parameters as in (12), the distribution of X_T is, in general, no longer available. This then complicates the task of computing θ (either analytically or by estimating it by simulating X_T *directly*).

One solution to this problem is to simulate X_T *indirectly* by *simulating the SDE* in (12). ■

Exercise 4 *Suppose we assume that the short-rate, r_t, has dynamics given by (12). What do the comments in the above example then imply about the difficulty of computing the term-structure?*

Exercise 5 *Suppose you wish to estimate $\theta := \mathrm{E}[f(\{X_t\}_{0\leq t\leq T})]$ so that $f(\cdot)$ now depends on the entire path of the process, X_t. Comment on whether or not this problem might be reduced to the problem of estimating $\theta = \mathrm{E}[f(Y_T)]$ for some process, Y_T.*

Remark 1 *The situation of Example 3 where we do not know the distribution of X_T is typical. As a result, it is often necessary to simulate a stochastic differential equation if we wish to estimate some associated quantity, e.g. $\theta = \mathrm{E}[f(X_T)]$. We describe how to do this below, beginning with the one-dimensional case.*

Simulating a 1-Dimensional SDE: The Euler Scheme

Let us assume that we are faced with an SDE of the form

$$dX_t = \mu(t, X_t)\;dt + \sigma(t, X_t)\;dB_t \tag{13}$$

and that we wish to simulate values of X_T but do not know its distribution. (This could be due to the fact that we cannot solve (13) to obtain an explicit solution for X_T, or because we simply cannot determine the distribution of X_T even though we do know how to solve (13)).

When we simulate an SDE, what we mean is that we simulate a *discretized* version of the SDE. In particular, we simulate a discretized process, $\{\widehat{X}_h, \widehat{X}_{2h},\; \ldots\;, \widehat{X}_{mh}\}$, where m is the number of time steps, h is a constant and $mh = T$. The smaller the value of h, the closer our discretized path will be to the continuous-time path we wish to simulate. Of course this will be at the expense of greater computational effort. While there are a number of

[1] As we did in Examples 1 and 2

discretization schemes available, we will focus on the simplest and perhaps most common scheme, the *Euler* scheme.

The Euler scheme is intuitive, easy to implement and satisfies

$$\widehat{X}_{kh} \;=\; \widehat{X}_{(k-1)h} + \mu\left((k-1)h,\ \widehat{X}_{(k-1)h}\right)\ h \;+\; \sigma\left((k-1)h,\ \widehat{X}_{(k-1)h}\right)\ \sqrt{h}\ Z_k \tag{14}$$

where the Z_k's are IID $N(0,1)$. If we want to estimate $\theta := \mathrm{E}[f(X_T)]$ using the Euler scheme, then for a fixed number of paths, n, and discretization interval, h, we have the following algorithm.

Using the Euler Scheme to Estimate $\theta = \mathrm{E}[f(X_T)]$
When X_t Follows a 1-Dimensional SDE

$t = 0$; $\widehat{X} = X_0$
for $j = 1$ **to** n
  for $k = 1$ **to** $T/h = m$
    generate $Z \sim N(0,1)$
    set $\widehat{X} = \widehat{X} + \mu(t, \widehat{X})h + \sigma(t, \widehat{X})\sqrt{h}\ Z$
    set $t = t + h$
  end for
  set $f_j = f(\widehat{X})$
end for
set $\widehat{\theta}_n = (f_1 + \ldots + f_n))/n$
set $\widehat{\sigma}_n^2 = \sum_{j=1}^n (f_j - \widehat{\theta}_n)^2/(n-1)$
set approx. $100(1-\alpha)$ % CI $= \widehat{\theta}_n \pm z_{1-\alpha/2}\frac{\widehat{\sigma}_n}{\sqrt{n}}$

Remark 2 *Observe that even though we only care about X_T, we still need to generate intermediate values, X_{ih}, if we are to minimize the discretization*[2] *error.*

Remark 3 *If we wished to estimate $\theta = \mathrm{E}[f(X_{t_1}, \ldots, X_{t_p})]$ then in general we would need to keep track of $(X_{t_1}, \ldots, X_{t_p})$ inside the inner for-loop of the algorithm.*

Exercise 6 *Can you think of a derivative where the payoff depends on $(X_{t_1}, \ldots, X_{t_p})$, but where it would* not *be necessary to keep track of $(X_{t_1}, \ldots, X_{t_p})$ on each sample path?*

Discretization Error

The *discretization error* may be defined by $D := |\mathrm{E}[f(X_T)] - \mathrm{E}[f(\widehat{X}_T)]|$ and it is very important when simulating SDE's to ensure that D is sufficiently small. Otherwise while our estimate, $\widehat{\theta}_n$, will be an unbiased estimate of $\mathrm{E}[f(\widehat{X}_T)]$, it will be a very biased estimate of $\mathrm{E}[f(X_T)]$, the quantity of interest.

A common method[3] of controlling discretization error is as follows. Let $\widehat{\theta}^{(m)}$ be our estimator of $\mathrm{E}[f(\widehat{X}_T)]$ when we use m discretization points. We first compute $\widehat{\theta}^{(m)}$ and $\widehat{\theta}^{(2m)}$ for a reasonably large[4] value of m. If $|\widehat{\theta}^{(m)} - \widehat{\theta}^{(2m)}|$ is sufficiently small then we can assume that $2m$ is a sufficiently large sample size to guarantee a

[2] We discuss discretization error in more detail below.
[3] This applies to both one-dimensional and multiple-dimensional SDE's.
[4] For example, we might take $m = 100$ if T corresponds to approximately 1 year.

sufficiently small discretization error. If $|\widehat{\theta}^{(m)} - \widehat{\theta}^{(2m)}|$ is not sufficiently small then we compute and compare $|\widehat{\theta}^{(4m)} - \widehat{\theta}^{(2m)}|$. We continue in this manner until we are satisfied with the size of D. Note that by controlling the discretization error in this manner, we have added substantially to the amount of computational time required.

Other similar methods are available more generally that also take into account the optimal tradeoff between the sample size, n, and the number of discretization points, m. A smaller value of m will result in greater discretization error, whereas a smaller value of n will result in greater statistical error. If there is a fixed computational budget then it is important to choose n and m in an optimal manner.

Simulating a Multidimensional SDE

In the multidimensional case, $\mathbf{X}_t$, $\mathbf{B}_t$ and $\mu(t, \mathbf{X}_t)$ in (13) are now vectors, and $\sigma(t, \mathbf{X_t})$ is a matrix. This situation arises when we have a *series* of SDE's in our model. This could occur in a number of financial engineering contexts. Some examples include:

(1) Modelling the evolution of multiple stocks. This might be necessary if we are trying to price derivatives whose values depend on multiple stocks or state variables, or if we are studying the properties of some portfolio strategy with multiple assets.

(2) Modelling the evolution of a single stock where we assume that the volatility of the stock is itself stochastic. Such a model is termed a *stochastic volatility* model.

(3) Modelling the evolution of interest rates. For example, if we assume that the short rate, r_t, is driven by a number of factors which themselves are stochastic and satisfy SDE's, then simulating r_t amounts to simulating the SDE's that drive the factors. Such models occur in short-rate models as well as HJM and Market LIBOR models.

In all of these cases, whether or not we will have to simulate the SDE's will depend on the model in question and on the particular quantity that we wish to compute. If we do need to discretize the SDE's and simulate their discretized versions, then it is very straightforward. If there are n correlated Brownian motions driving the SDE's, then at each time step, t_i, we must generate n IID $N(0,1)$ random variables. We would then use the Cholesky Decomposition to generate $X_{t_{i+1}}$. This is exactly analogous to our method of generating correlated geometric Brownian motions. In the context of simulating multidimensional SDE's, however, it is more common to use independent Brownian motions as any correlations between components of the vector, $\mathbf{X}_t$, can be induced through the matrix, $\sigma(t, \mathbf{X}_t)$.

Remark 4 *There are other important issues that arise when simulating SDE's. For example, while we have only described the Euler scheme, there are other more sophisticated discretization schemes that can also be used. In a sense that we will not define, these schemes have superior convergence properties than the Euler scheme. However, they are sometimes more difficult to implement, particularly in the multi-dimensional setting.*

---

3 Applications to Financial Engineering

Example 4 (Option Pricing Under Stochastic Volatility)

Suppose the evolution of the stock price, S_t, under the risk-neutral probability measure is given by

$$dS_t = rS_t\ dt + \sqrt{V_t} S_t\ dB_t^{(1)} \tag{15}$$

$$dV_t = \alpha\,(b - V_t)\ dt + \sigma\sqrt{V_t}\ dB_t^{(2)}. \tag{16}$$

If we want to price a European call option on the stock with expiration, T, and strike K, then the price is given by

$$C_0 = \exp(-rT)\mathrm{E}[\max(S_T - K,\ 0)].$$

We could estimate C_0 by simulating n sample paths of $\{S_t, V_t\}$ up to time T, and taking the average of $\exp(-rT)\max(S_T - K,\ 0)$ over the n paths as our estimated call option price, $\widehat{C}_0$. ∎

Exercise 7 *Write out the details of the algorithm that you would use to estimate C_0 in Example 4.*

Example 5 (Portfolio Evaluation)

Suppose an investor trades continuously in a particular fund whose time t value is denoted by P_t. Any cash that is not invested in the fund earns interest in a cash account at the risk-free rate, r_t. Assume that the dynamics of P_t are given by

$$\begin{aligned} dP_t &= P_t \left[\ (\mu + \lambda X_t)\ dt + \sigma_1\ dB_t^{(1)} + \sigma_2\ dB_t^{(2)}\right] \\ dX_t &= -kX_t\ dt + \sigma_{x,1}\ dB_t^{(1)} + \sigma_{x,2}\ dB_t^{(2)} \\ dr_t &= \alpha(t, r_t)\ dt + \beta(t, r_t)\ dB_t^{(3)} \end{aligned}$$

where X_t is a *state variable* that possibly represents the time t value of some relevant economic variable. Let θ_t be a predictable process that denotes the *fraction* of the investor's wealth that is invested[5] in the fund at time t, and let W_t denote the investor's wealth at time t. We then see that W_t satisfies

$$dW_t \ = \ [r_t + \theta_t(\mu + \lambda X_t - r)]W_t\ dt \ + \ \theta_t W_t[\sigma_1\ dB_t^{(1)} + \sigma_2\ dB_t^{(2)}]. \tag{17}$$

Now it may be the case that the investor wishes to compute $\mathrm{E}[u(W_T)]$ where $u(\cdot)$ is his utility function, or that he wishes to compute $\mathbf{P}(W_T \leq a)$ for some fixed value, a. In general, however, it is not possible to perform these computations explicitly. As a result, we could instead use simulation. Noting that we will not in general be able to solve (17) for W_T and its distribution, this means that we would have to simulate the multivariate SDE satisfied by (P_t, X_t, r_t, W_t) in order to answer these questions. ∎

Exercise 8 *Write out the details of the algorithm that you would use to estimate $\theta = \mathbf{P}(W_T \leq a)$ in Example 5.*

Example 6 (The CIR Model with Time-Dependent Parameters)

We assume the Q-dynamics of the short-rate, r_t, are given by

$$dr_t \ = \ \alpha[\mu(t) - r_t]\ dt \ + \ \sigma\sqrt{r_t}\ dB_t \tag{18}$$

where $\mu(t)$ is a deterministic function of time. This generalized CIR model is used when we want to fit a CIR-type model to the initial term-structure.

Suppose now that we wish to price a derivative security maturing at time T with payoff $C_T(r_T)$. Then its time 0 price, C_0, is given by

$$C_0 = \mathrm{E}_0\left[e^{-\int_0^T r_s\ ds}\ C_T(r_T)\right]. \tag{19}$$

The distribution of r_t is not available in an easy-to-use closed form so perhaps the easiest way to estimate C_0 is by simulating the dynamics of r_t. Towards this end, we could either use (18) and simulate r_t directly or

[5]This means that $1 - \theta_t$ is invested in the cash account at time t.

alternatively, we could simulate $X_t := f(r_t)$ where $f(\cdot)$ is an invertible transformation. Note that because of the discount factor in (19), it is also necessary to simulate the process, Y_t, given by

$$Y_t = \exp\left(-\int_0^t r_s \, ds\right).$$

▮

Exercise 9 *Describe in detail how you would you would estimate C_0 in Example 6. Note that there are alternative ways to do this. What way do you prefer?*

Variance Reduction Methods

Simulating SDE's is a computationally intensive task as we need to do a lot of work for each sample that we generate. As a result, variance reduction techniques are often very useful in such contexts. For an example based on conditional Monte-Carlo, consider again the stochastic volatility model of Example 4.

Example 7 (Conditional Monte-Carlo for the Stochastic Volatility Model)

As before, we will use $c(x, t, K, r, \sigma)$ to denote the Black-Scholes price of a European call option when the current stock price is x, the time to maturity is t, the strike is K, the risk-free interest rate is r and the volatility is σ.

We will also use the following fact: if v_t is a deterministic function of time, then

$$\int_0^T v_t \, dB_t \sim \mathsf{N}\left(0, \int_0^T v_t^2 \, dt\right).$$

Suppose now that the Brownian motions, $B_t^{(1)}$ and $B_t^{(2)}$ in (15) and (16), are independent. Then

$$C_0 = e^{-rT} \, \mathrm{E}[\max(S_T - K, 0)] = e^{-rT} \, \mathrm{E}\left[\mathrm{E}[\max(S_T - K, 0) \mid V_t, \; 0 \le t \le T]\right].$$

But it can be shown using the independence of $B_t^{(1)}$ and $B_t^{(2)}$ that

$$e^{-rT} \, \mathrm{E}[\max(S_T - K, 0) \mid V_t, \; 0 \le t \le T] = c(S_0, T, K, r, V)$$

where $V := \sqrt{\int_0^T V_t \, dt / T}$. In particular, this means that we can estimate C_0 by using conditional Monte-Carlo method. ▮

Exercise 10 *Write out the details of the conditional Monte-Carlo algorithm that you would use to estimate C_0.*

Remark 5 *The above example may be generalized in certain circumstances to accommodate dependence between $B_t^{(1)}$ and $B_t^{(2)}$.*

4 Simulating Jump-Diffusion Models

Let S_t denote the time t price of a security and assume that the risk-neutral dynamics of S_t are given by

$$S_t = S_0 \, e^{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B_t} \prod_{i=1}^{N_t} Y_i \tag{20}$$

where B_t is a standard Brownian motion, N_t is Poisson process with intensity λ that is independent of B_t, and the Y_i's are a sequence of IID random variables that are independent of both B_t and N_t. We know that under the risk-neutral probability measure, Q, it must be the case that $M_t := S_t \exp(-rt)$ is a martingale. This then implies $\mu = r - \lambda(\mu_y - 1)$ where $\mu_y = \mathrm{E}[Y_i]$.

Exercise 11 *Confirm the expression for μ_y above by showing that*

$$\mathrm{E}\left[S_t \exp(-rt)\right] = S_0 e^{[\mu - r + \lambda(\mu_y - 1)]t}.$$

Note that the paths of the security price are no longer *continuous* due to the presence of jumps. We therefore have to be careful when we define S_t. In particular, we use the standard convention that

$$S_t := \lim_{u \downarrow t} S_u$$

so that S_t is *right-continuous*. We use S_{t-} to denote the left-limit so that

$$S_{t-} := \lim_{u \uparrow t} S_u.$$

Of course $S_t = S_{t-}$ if a jump does not take place at time t. We may also express[6] equation (20) in differential notation and obtain

$$dS_t = \mu S_{t-} \, dt + \sigma S_{t-} \, dW_t + S_{t-} \, dJ_t \tag{21}$$

where $J_t := \sum_{j=1}^{N_t} (Y_j - 1)$. For obvious reasons, this model is a particular example of what are called *jump-diffusion* models. Jump-diffusion models are often expressed as stochastic differential equations as is the case in (21).

Exercise 12 *Show that equation (21) implies that $S_{\tau_i} = S_{\tau_i -} Y_i$ where τ_i is the i^{th} jump time.*

Exercise 13 *What condition is required of the Y_i's to ensure limited liability of the shareholder, i.e., to ensure that S_t can never go negative?*

For the particular jump-diffusion model in (20) (or equivalently, (21)), it is clear that the security price (under Q) behaves like a geometric Brownian motion in between jumps. As long as we can simulate samples of Y, it is therefore easy to generate samples of S_T.

Exercise 14 *How would you simulate $S_{t_1}, \ldots, S_{t_m}$, where $t_1 < t_2 \ldots < t_m$ are fixed times?*

Exercise 15 *How would you simulate $S_{\tau_1}, \ldots, S_{\tau_{N_T}}$, where τ_i is the i^{th} jump time of the security price?*

Exercise 16 *If N_t was a non-homogeneous Poisson process, could we still exactly simulate samples from the jump-diffusion model?*

Remark 6 *Merton (1976) was the first to introduce the jump-diffusion model of (20). In his model, the Y_i's were lognormally distributed. This assumption simplified (why?) the computation of European option prices.*

[6] Some work is actually required to show that (20) is indeed the solution to (21). We will not concern ourselves with how to do this in this course.

Simulating Stochastic Differential Equations: Jump-Diffusion Models

We now briefly discuss how to approximately simulate certain types of jump-diffusion processes when exact simulation is impossible. As in Section 2, we discretize time and utilize an Euler-type scheme. Let N_t be a Poisson process, B_t a standard Brownian motion and $Y := \{Y_1, Y_2, \ldots\}$ a sequence of IID random variables. We assume N_t, B_t and Y are all independent of one another. Consider now a jump-diffusion model of the form

$$dX_t \;=\; \mu(X_{t-})\; dt + \sigma(X_{t-})\; dB_t + c(X_{t-}, Y_{y_{N_{t-}+1}})\; dN_t. \tag{22}$$

X_t might represent the time t value of an underlying security or relevant state variable. We can approximately simulate a path of X_t on $[0, T]$ in a number of ways. We now describe one such way:

1. Define an initial grid $0, h, 2h, \ldots, mh = T$. This is what we did in Section 2.
2. Since the Poisson process, N_t, is independent of W_t, we can imagine that we first simulate the jump times of the process in $[0, T]$. Let these times be $\tau_1, \ldots, \tau_{N_T}$, noting of course that N_T will vary from sample path to sample path.
3. We now create a combined time grid, $0 = t_0, t_1, \ldots, t_M = T$ consisting of the original $mh + 1$ grid points as well as the N_T jump times. We therefore have $M = mh + 1 + N_T$.
4. We then approximately simulate X_t at points on the combined grid.

Exercise 17 *How would you perform step* 4 *above?*

If we want to generate n sample paths then we repeat steps 1 to 4 a total of n times, noting that we obtain a different combined grid for each sample path.

It is possible to construct many other types of jump-diffusion models. For example, instead of using a Poisson process we can use more general *point processes* that have *stochastic intensities*. We can of course also construct high-dimensional processes with multiple Brownian motions and point processes driving the dynamics of the state variables. When it comes to simulating these processes it is generally necessary to use a discretization scheme. Many of these discretization schemes are easy to construct and make intuitive sense, as for example the Euler scheme does for diffusions. Understanding the convergence properties of these schemes, however, is technically demanding and is an area of current research. Some details, along with further references, can be found in Glasserman (2003).

References

Kloeden, P., and E. Platen. 1992. *Numerical Solution of Stochastic Differential Equations*. Berlin: Springer-Verlag.

Glasserman, P. 2003. *Monte Carlo Methods in Financial Engineering.* Springer-Verlag, New York.