Chapter 3 Monte Carlo Simulation with Stochastic Differential Equations

Fig. 3.1. Illustration of the Monte Carlo approach for a European put, with $K = 50$, $S_0 = 50$, $T = 1$, $\sigma = 0.2$, $r = 0$; five simulated paths in the $(S, t)$-plane with payoff; vertical axis: $V$. The front curve $V(S, 0)$ is shown.

The Sections 1.5 and 1.7.3 have introduced the principle of risk-neutral evaluation, which can be summarized by

$$V(S_0, 0) = e^{-rT}E_Q(V(S_T, T) \mid S_t \text{ starting from } (S_0, 0)),$$

where $E_Q$ represents the expectation under a risk-neutral measure. For the Black–Scholes model, this expectation is an integral as in (1.50). This suggests two approaches of calculating $V$. Either approximate the integral, or calculate the expectation by simulating the underlying stochastic differential equation (SDE) repeatedly. The latter approach is illustrated in Figure 3.1. Five paths $S_t$ are calculated for $0 \leq t \leq T$ in the risk-neutral fashion, each starting from $S_0$. Then for each resulting $S_T$ the payoff is calculated, here for a European put. The figure illustrates the bulk of the work. (In reality, thousands of paths are calculated.) It remains the comparably cheap task of calculating the mean of the payoffs as approximation for $E_Q$. This is the Monte Carlo approach. The Monte Carlo approach works for general models, for example, for systems of equations, see Figure 3.2.
This chapter is based on the ability to numerically integrate SDEs. Therefore a significant part of the chapter is devoted to this topic. Again \( X_t \) denotes a stochastic process and a solution of an SDE (1.31),

\[
dX_t = a(X_t, t) \, dt + b(X_t, t) \, dW_t \quad \text{for } 0 \leq t \leq T,
\]

where the driving process \( W \) is a Wiener process. We assume a \( t \)-grid with \( 0 = t_0 < t_1 < \ldots \). For convenience, the step length \( \Delta t = t_{j+1} - t_j \) is taken equidistant. As is common usage in numerical analysis, we also use the \( h \)-notation, \( h := \Delta t \). For \( \Delta t = h = T/m \) the index \( j \) runs from 0 to \( m - 1 \). The solution of a discrete version of the SDE is denoted \( y_j \). That is, \( y_j \) should be an approximation to \( X_{t_j} \), or \( y_t \) an approximation to \( X_t \). Weaker requirements will be discussed below. The initial value for \( t = 0 \) is assumed a given constant,

\[ y_0 = X_0. \]

For example, from Algorithm 1.11 we know the Euler discretization

\[
\begin{align*}
y_{j+1} &= y_j + a(y_j, t_j) \Delta t + b(y_j, t_j) \Delta W_j, \\
\Delta W_j &= W_{t_{j+1}} - W_{t_j} = Z \sqrt{\Delta t} \quad \text{with } Z \sim N(0,1).
\end{align*}
\]

Since an approximation \( y_T \) also depends on the chosen step length \( h \), we also write \( y_T^h \). From numerical methods for deterministic ODEs (\( b \equiv 0 \)) we know the discretization error of Euler’s method is \( O(h) \),

\[ X_T - y_T^h = O(h). \]

The Algorithm 1.11 (repeated in equation (3.1)) is an explicit method in that in every step \( j \to j+1 \) the values of the functions \( a \) and \( b \) are evaluated at the previous approximation \( (y_j, t_j) \). Evaluating \( b \) at the left-hand mesh point \( (y_j, t_j) \) is consistent with the Itô integral and the Itô process, compare the notes at the end of Chapter 1.

After we have seen in Chapter 2 how \( Z \sim N(0,1) \) can be calculated, all elements of Algorithm 1.11 are known, and we are equipped with a method to numerically integrate SDEs (\( \rightarrow \) Exercise 3.1). In this chapter we learn about other methods, and discuss the accuracy of numerical solutions of SDEs. The exposition of Sections 3.1 through 3.3 follows [KLP92]. Readers content with Euler’s method (3.1) may like to skip these sections.

After a brief exposition on constructing bridges (Section 3.4), we turn to the main theme, namely, Monte Carlo methods for pricing options. The basic principle is outlined for European options (Section 3.5). For American options parametric methods and regression methods are introduced in Section 3.6. The final Section 3.7 discusses the calculation of sensitivities.