CS522 - Exotic and Path-Dependent Options

Tibor Jánosi

May 5, 2005

0.1 Other Option Types

We have studied extensively European and American puts and calls. The class of options is much larger, however.

A **digital (binary) option** is an option that has a discontinuous payoff. An example of such an option is the European cash-or-nothing option, whose payoff at the expiration time T is given by the following formula:

$$V_B(T,S) = \begin{cases} 1, \ if \ S > K \\ 0, \ otherwise \end{cases} .$$

As usual, we have denoted the strike price by K.

An **Asian option** is an option whose payoff depends on a suitably defined average $\bar{S}$ of the underlying stock's price between the inception of the option and its expiration. The payoff of an Asian call with strike K is then given by

$$C_A(T,\bar{S}) = \max(\bar{S} - K, 0).$$

Note that the second argument of C in the formula above is not the final stock price, but its average. This average can be computed continuously, or at discrete points only:

$$\begin{aligned} \bar{S} &= \frac{1}{T}\int_0^T S(t)dt \\ \bar{S} &= \frac{1}{n}\sum_{i=1}^{n} S(t_i), \text{ where } 0 \leqslant t_1 < t_2 < \ldots < t_n \leqslant T \end{aligned}$$

A **fixed lookback** call or put is an option whose value depends on the maximum or the minimum of the underlying's price during the lifetime of the option, respectively:

$$\begin{aligned} C_L^{fixed}(T) &= \max(\max_{t\in[0,T]} S(t) - K, 0) \\ P_L^{fixed}(T) &= \max(K - \min_{t\in[0,T]} S(t), 0) \end{aligned}$$

Like in the case of Asian options, the maximum or the minimum stock price can be computed by sampling the stock price at discrete moments in time.

In the case of a **floating lookback** option the strike price is not fixed, but it is chosen based on the best available price of the underlying over the lifetime of the option:

$$\begin{aligned} C_L^{\text{float}}(T) &= \max(S - \min_{t\in[0,T]} S(t), 0) \\ P_L^{\text{float}}(T) &= \max(\max_{t\in[0,T]} S(t) - S, 0) \end{aligned}$$

A large class of options is that of **barrier** options. Barrier options have knock-in and/or knock-out features. A knock-in feature is a condition that causes the option to be effective only if the underlying reaches a certain price level. A knock-out feature is a condition that causes the option to terminate immediately if the underlying reaches a certain price level.

There are four main types of barrier options: "up-and-in", "up-and-out", "down-and-in", and "down-and-out."

The "up-and-in" option becomes effective only if the underlying first reaches a predetermined knock-in barrier from below. Once the barrier is reached, the option behaves like a plain-vanilla option.

The "down-and-out" option behaves like a plain vanilla option as long as the knock-out-barrier specified for the respective option is not reached from above. When - and if - that happens, the option immediately terminates.

The definition of the "up-and-out" and "down-and-in" options is analogous.

Given that we have four types of barrier options, and that they can be either puts and calls with European or American features, we have implicitly defined 16 barrier option types, each with its own valuation problem.

Even more complicated options can be defined. For example, one can define options that knock-in only if the price of the underlying enters and stays inside a band of prices delimited by an upper and lower barrier.

It is not necessary for the underlying of an option to be a stock. One can define options on options, leading to **compound options**, e.g. a call on a put. A **chooser** option is an option expiring at time T_1 which allows the holder to choose between a plain-vanilla put or a call expiring at time $T_2 > T_1$. Options can be defined on underlying commodities (e.g. options on gold futures).

As you can see, there are many types of options, and there is almost a limitless theoretical possibility of adding new features. In practice, however, the choice of features is guided by the needs of the market. Options with features desired by many market participants are often traded on exchanges. Typically, these are simple options that have been widely studied theoretically and are well understood empirically. Complicated options, with many unusual features are often written by big financial companies at the request of their clients. Often, such values are hard to value, especially since they are illiquid.

0.2 Valuing Path-Dependent and Exotic Options

It is sometimes possible to write closed-form formulas for the value of exotic and path-dependent options.

0.2.1 Analytic Valuation: Chooser Options

Assuming that the money market account earns interest at the constant rate r, the time-t value of a chooser option that gives its holder the right to choose at time time T_1 between

a European put or a call with strike K and expiration date $T_2 \geqslant T_1$ is given by the following formula:

$$V(t, T_1, T_2; K) = p(t, T_2; K) + c(t, T_1; Ke^{-r(T_2-T_1)}),$$

i.e. the value of chooser option is equal to the value of a European put with expiration date T_2 and strike price K, plus the value of a European call with expiration date T_1 and strike price $Ke^{-r(T_2-T_1)}$. We provide the proof of this claim below.

When we are at time T_1, and we hold a chooser option, the rational decision is to choose the underlying plain-vanilla put or call with the higher value. For any underlying stock price $S(T)$, we will thus choose the call if $c(T_1, T_2; K) > p(T_1, T_2; K)$; if not, we will choose the put. Let us denote by A the set of all states of the world at time T_1 for which the underlying call is more valuable than underlying the put. Let function $\mathbb{I}_A$ be the indicator function of set A. Since all the relevant parameters (r, σ, T_1, T_2, K) are fixed, in our model the states of the world at time T_1 are distinguished only by the price of the stock price at that very same moment of time T_1. To avoid complicating our notation further, we identify these states with their associated stock price.

The time-T_2 payoff of the chooser option[1] is given by the following formula:

$$V(T_2, T_1, T_2; K) = \begin{cases} \max(S(T_2) - K, 0), \text{ if } \mathbb{I}_A(S(T_1)) = 1 \\ \max(K - S(T_2), 0), \text{ if } \mathbb{I}_A(S(T_1)) = 0 \end{cases}.$$

The notation for the value of the chooser option is somewhat unwieldy, but informative. $V(t_a, t_b, t_c; K)$ represents the time-t_a value of a chooser option expiring at time t_b; the put or call that the chooser option make available will expire at time t_c and their common strike price is K.

The formula expresses the very simple fact that if at time T_1 we went through a state in set A, then we chose the call, and we will receive its payoff. If we went through a state that is not in A (i.e. we went though a state in the complement of A, A^C), then we will receive the payoff of the put. By introducing the notation $(\bullet)^+ = \max(\bullet, 0)$ and the indicator function $\mathbb{I}_{A^C}$ of the set A^C, we get

$$V(T_2, T_1, T_2; K) = (S(T_2) - K)^+ \mathbb{I}_A(S(T_1)) + (K - S(T_2))^+ \mathbb{I}_{A^C}(S(T_2)).$$

Since A and A^C have no common elements, and their union is the set of all possible states at time T_1, we can conclude that $\mathbb{I}_A(S(T_1)) + \mathbb{I}_{A^C}(S(T_1)) = 1$, irrespective of the value of $S(T_1)$. The time-T_2 of the chooser option can now be rewritten as follows:

$$\begin{aligned} V(T_2, T_1, T_2; K) &= (S(T_2) - K)^+ \mathbb{I}_A(S(T_1)) + (K - S(T_2))^+ (1 - \mathbb{I}_A(S(T_2))) \\ &= \left[(S(T_2) - K)^+ - (K - S(T_2))^+\right] \mathbb{I}_A(S(T_1)) + (K - S(T_2))^+ \\ &= (S(T_2) - K) \mathbb{I}_A(S(T_1)) + (K - S(T_2))^+ . \end{aligned}$$

[1] Strictly speaking, we do not have a chooser option at T_2 anymore, since it expired at time T_1. We are really talking about the payoff of the instrument we got when we exercised the chooser option. For simplicity, however, we will keep talking about the chooser option's payoff at T_2.

The last equality has been obtained by noticing that the sum $(S(T_2)-K)^+-(K-S(T_2))^+$ represents the payoff of a portfolio consisting of a long call and a short put. An application of the put-call parity for European options expiring at time T_2 immediately produces the equivalent expression.

Now that we have an analytic expression for the value of the chooser option (more precisely, whatever became of it) at time T_2, we can compute its value at time T_1 by determining the discounted expect value of the payoff under the equivalent martingale probability measure:

$$\begin{aligned}
V(T_1,T_1,T_2;K) &= e^{-r(T_2-T_1)}\mathbb{E}_q\left[(S(T_2)-K)\,\mathbb{I}_A(S(T_1))+(K-S(T_2))^+\right] \\
&= e^{-r(T_2-T_1)}\mathbb{I}_A(S(T_1))\mathbb{E}_q\left[(S(T_2)-K)\right]+e^{-r(T_2-T_1)}\mathbb{E}_q\left[(K-S(T_2))^+\right] \\
&= e^{-r(T_2-T_1)}\mathbb{I}_A(S(T_1))\mathbb{E}_q\left[(S(T_2)-K)\right]+p(T_1,T_2;K) \\
&= \mathbb{I}_A(S(T_1))\left(e^{-r(T_2-T_1)}\mathbb{E}_q\left[S(T_2)\right]-Ke^{-r(T_2-T_1)}\right)+p(T_1,T_2;K) \\
&= \mathbb{I}_A(S(T_1))\left(S(T_1)-Ke^{-r(T_2-T_1)}\right)+p(T_1,T_2;K).
\end{aligned}$$

The first term in the expression above has been rewritten by first removing the indicator function from under the expectation operator. This step is justified by noting that at time T_1 the value of the underlying stock $S(T_1)$, and thus the value of the indicator function has been totally determined; this quantity is not random from time T_1 on. Noting that the expectation of a constant (K) is the constant itself, and that the discounted expected stock $S(T_2)$ is equal to the stock price at time T_1 (why?), we can completely eliminate the expectation operator.

The second term corresponds to the definition of the time-T_1 value of a European call with expiration T_2 and strike price K.

Let us now revisit the condition that defines states in set A:

$$\begin{gathered}
c(T_1,T_2;K)>p(T_1,T_2;K) \\
c(T_1,T_2;K)-p(T_1,T_2;K)>0 \\
S(T_1)-Ke^{-r(T_2-T_1)}>0
\end{gathered}$$

The last relation has been obtained by using the put-call parity for European puts.

Note that the expression that defines the set of states A is the same as the expression multiplying the indicator function in the expression for $V(T_1,T_1,T_2;K)$. By observing that

$$\mathbb{I}_A(S(T_1))\left(S(T_1)-Ke^{-r(T_2-T_1)}\right)=\begin{cases} S(T_1)-Ke^{-r(T_2-T_1)}, \text{ if } S(T_1)-Ke^{-r(T_2-T_1)}>0 \\ 0, \text{ otherwise}\end{cases},$$

we can rewrite the leftmost term in $V(T_1,T_1,T_2;K)$ as $\left(S(T_1)-Ke^{-r(T_2-T_1)}\right)^+$. We now get:

$$\begin{aligned}
V(T_1,T_1,T_2;K) &= \left(S(T_1)-Ke^{-r(T_2-T_1)}\right)^+ + p(T_1,T_2;K) \\
&= c(T_1,T_1;Ke^{-r(T_2-T_1)})+p(T_1,T_2;K).
\end{aligned}$$

Thus, at time T_1 the value of the chooser option is given by the value of a European call with expiration T_1 and strike $Ke^{-r(T_2-T_1)}$, plus the value of a European put expiring at T_2 with strike price K. A simple arbitrage argument immediately leads to the conclusion that the equality must hold for all times t between 0 and T_1:

$$V(t, T_1, T_2; K) = c(t, T_1; Ke^{-r(T_2-T_1)}) + p(t, T_2; K).$$

A chooser option appears to be, and in some sense is, a complicated instrument. Careful reasoning and the knowledge we gained from studying plain-vanilla options allowed us to decompose the chooser option into a portfolio of one European put and one European call. These instruments, in turn, can be valued using the Black-Scholes formulas.

It is often possible to decompose complex instruments (not only options) into simpler ones. Sometimes, like here, the valuation of the simpler instruments is straightforward, but this is not always the case.

Can you derive an alternative valuation formula for chooser options? Hint: Express the value of $\mathbb{I}_A$ as a function of $\mathbb{I}_{A^C}$.

0.2.2 Binomial Method: General, but Slow

If an instrument can not be valued using closed-form formulas, then one must look for numerical alternatives. As we have seen earlier, approximate values for simple options are easy to compute using the binomial method. We have also seen, however, that certain features of the underlying option make the binomial method to become resource intensive, when analyzed from an algorithmic perspective.

In particular, we have shown that if a stock pays discrete dividends, then each node that corresponds to a given dividend payment "sprouts" a new subtree whose nodes will not recombine with the nodes of subtrees associated with its neighboring nodes. This yields to an exponential increase in the total number of nodes in a binomial tree, when viewed as a function of the number of dividend payments. The presence of discrete dividends can greatly increase the running time and memory consumption needed for evaluating even simple options. While we did not address this topic extensively, it should be clear to an informed reader that discrete dividends will complicate valuation methods based on differential equations or inequalities as well (they induce discontinuities in the solution).

Path dependent options impose another type of complexity, which also yields to an exponential increase in the number of states. Let us consider an European instrument whose payoff is an arbitrary function of all the stock prices encountered on the specific path followed from time 0 to expiration. This means that each individual node in which we could end up at expiration must be associated with the entire history of stock price evolution.

There are several ways in which one could achieve this association. The easiest one, and the one typically given in textbooks, is to not to recombine the nodes. If this approach is followed, then each node is associated with a unique path from the root of the tree (time-0) to the respective node. Hence, knowing what node we are in allows us to determine

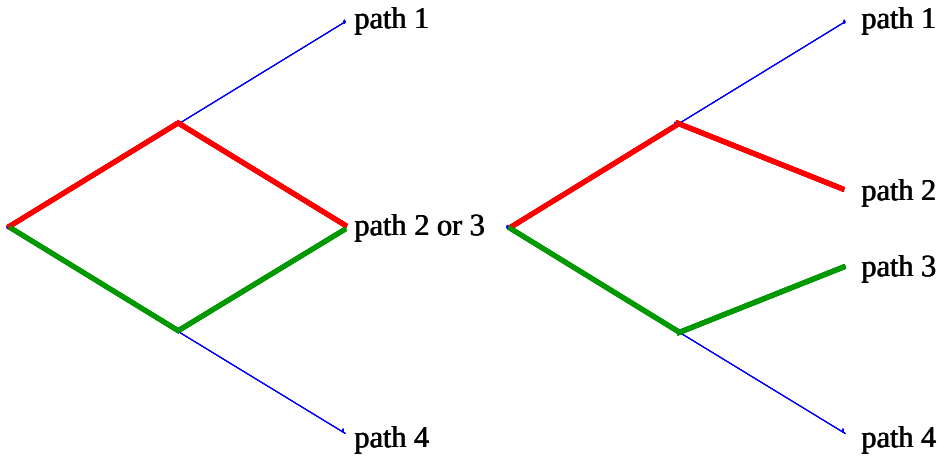

Figure 1: Tracking paths in binomial trees with recombining and non-recombining nodes.

fully the price history of the underlying instrument, as illustrated in figure 1. This idea is easy to implement, and very general, but it results in the doubling of the number of nodes at each level of the binomial tree. Such a binomial tree that contains n intervals (thus $n+1$ levels) will have 2^n nodes on the last level, and a total of $2^{n+1}-1$ nodes. The desire to use high values for n, which is necessary to increase precision, thus conflicts with the resource-consumption limits imposed by the size of the tree.

From an algorithmic perspective, it is not strictly necessary to use a non-recombining tree in order to maintain the price history of the underlying stock. In principle, it is possible to design data structures that allow the user to maintain in each node a list of all possible incoming paths, and compute and store separately all the relevant values that are associated with those paths. Such solutions, while logically equivalent to a binomial tree, are complex to implement and non-uniform. In the non-recombining tree each node can be reached only through one path from the root, and only only one value (set of values) characterizing that path must be accounted for. In our alternative solution, nodes far from the root can be reached through many paths, which progressively complicates accounting for the paths, and for the values associated with them. A more careful reasoning will show that the overall complexity and resource requirements of such alternative methods are not less than those of the corresponding simple non-recombining binomial tree. There are no shortcuts if the entire price history must be accounted for.

From a programmer's perspective, one big advantage of the binomial methods is that the valuation of many different instruments can be implemented in a very uniform manner. It is possible to write very general valuation algorithms based on the binomial method such that all the specific functionality is factored out into user-defined functions. You have solved a problem of this flavor when you built a general valuation function on top of a recombining binomial tree.

Of course, efficiency matters, and one is thus motivated to look for more efficient

alternatives.

Consider, for example, the case of Asian options with discrete price averaging. At expiration, the only prices that matter are the values sampled at times t_i, $i = \overline{1,n}$; prices at intermediate times are not relevant. As long as we associate each node at expiration with its own sequence of sampled prices, our binomial model will be useful for valuation. Indeed, we can solve this problem in a manner analogous to the treatment of discrete dividend payments, by "sprouting" a new independent subtree at each node that occurs at any of the sampling times t_i (and making sure that nodes from different subtrees never recombine). In-between sampling times, as well as between time 0 and the first sampling time, the subtrees will be recombining. Can you say why?

While the total number of nodes needed to implement this variation of the binomial model will still be exponential in the number of sampling times, avoiding the doubling of the number of nodes at each time step is can lead to a huge decrease in the number of nodes needed if there are many more time intervals represented in the tree than sampling times.

0.2.3 Numerical Solutions to Differential Equations and Inequalities

The solution of many valuation problems can be stated in terms of the solution of a problem involving (a system of) differential equations and inequalities with suitable initial, final, and boundary conditions. Some of these problems can be solved analytically, but many can only be computed numerically. Many mathematical problems that arise in the context of valuation have been studied extensively in other fields, such as physics and engineering. We do not expand on this topic further.

0.2.4 Monte Carlo Methods

There are no closed-form formulas for valuing discretely sampled Asian options. We will now study these options to gain insights into a new valuation technique, the Monte-Carlo method. As usual, we assume that the stock price follows a log-normal distribution. For concreteness, we will focus on Asian calls. The treatment of Asian puts is analogous.

The time-t value of the option is, as usual, the discounted expect value of the pay-off at expiration, where the expectation is computed under the equivalent martingale probabilities q:

$$V(t,T;K) = e^{-r(T-t)}\mathbb{E}_q\left[\max(\bar{S}-K,0)\right].$$

Let us assume for the moment that we have a general - but approximate - method of numerically estimating expectations $\mathbb{E}_q[\bullet]$.

One is often not interested only in determining the value of a particular instrument, but also on simultaneously determining the sensitivity of the respective value to changes in various underlying parameters. Knowing the value of an instrument is important for trading and accounting purposes, knowing its sensitivities is important for hedging.

Assume that we can compute the approximate value of an Asian option, and that the respective value is $\tilde{V}(t,T;K,p)$. Note that we have modified the notation to make

explicit the dependence of the value on some underlying parameter p. We can estimate the sensitivity of $\tilde{V}$ to small changes in the parameter p by using any of the numerical methods we have introduced for computing derivatives. Here is a method using symmetric differences:

$$\frac{\partial \tilde{V}}{\partial p}(t,T;K,p) \approx \frac{\tilde{V}(t,T;K,p+\delta p) - \tilde{V}(t,T;K,p)}{\delta p}.$$

Such an approach *might* have the downside that the valuation method for the respective option must be used twice in order to compute the desired sensitivity. We point out, however, that this is not necessarily the case, as some methods might generate values for an entire range of parameters. Think, for example, of the valuation problem for European or American options using finite differences, where we simultaneously compute the values of the option for an many points between $S_{\min}$ and S. Depending on the step size of the grid, one might be able to use these values to estimate the delta of the respective option.

0.2.5 Sensitivities of Asian Options as Expectations

Since we have an exact expression giving us the value of the Asian option as an expectation, we can, in principle, compute the sensitivity with respect to infinitesimal changes in the value of parameter p:

$$\begin{aligned}\frac{\partial V}{\partial p} &= \frac{\partial}{\partial p}\left[e^{-r(T-t)}\mathbb{E}_q\left[\max(\bar{S}-K,0)\right]\right] \\ &= \frac{\partial}{\partial p}\left[e^{-r(T-t)}\right]\mathbb{E}_q\left[\max(\bar{S}-K,0)\right] + e^{-r(T-t)}\frac{\partial}{\partial p}\mathbb{E}_q\left[\max(\bar{S}-K,0)\right].\end{aligned}$$

We assumed that we have a method of computing numerical values for computing expectations under q, but that does not imply that we know how to compute expressions like $\frac{\partial}{\partial p}\mathbb{E}_q[\bullet]$. Given our assumptions, it would be practical if we could interchange the order of the differential and expectation operator.

Do the Expectation and Differential Operator Commute? Is this possible? If yes, the following equality should hold in general, irrespective of the expression replaced by $\bullet$:

$$\frac{\partial}{\partial p}\mathbb{E}_q[\bullet] = \mathbb{E}_q\left[\frac{\partial}{\partial p}\bullet\right].$$

To make things more specific, let us examine the validity of this purported equality by using a digital "all-or-nothing" option. Let us consider one of the infinitely many possible paths that the stock price might follow between time 0 and the expiration time T. At expiration, the payoff of the digital option is differentiable for all underlying stock prices, except at $S(T) = K$. It is easy to see that irrespective of the parameter p at hand, we have that $\frac{\partial V_{digital}}{\partial p}(T,T;K) = 0$ for all $S \neq K$, i.e. $\frac{\partial V_{digital}}{\partial p}(T,T;K) = 0$ a.s. ("almost surely"). From this, we immediately conclude that $\mathbb{E}_q\left[\frac{\partial V_{digital}}{\partial p}(T,T;K)\right] = 0$. If

the differential and expectation operator were commutable, the last relation would imply that $\frac{\partial}{\partial p}\mathbb{E}_q\left[V_{digital}(T,T;K)\right]=0$, which, in turn, means that

$$\frac{\partial V_{digital}}{\partial p}(t,T;K)=\frac{\partial}{\partial p}\left[e^{-r(T-t)}\right]\mathbb{E}_q\left[V_{digital}(T,T;K)\right].$$

It is easy to see why this equality can not be true for all parameters p. The last relation implies that the value of a digital option is not sensitive to changes in the volatility of the underlying stock (σ), or the stock price at time t. This can not be the case.

Thus, in general, the differential and the expectation operator do not commute. With this caveat, we note that it is possible for the two operators to commute, assuming that the function represented by $\bullet$ above is uniformly integrable. In the following, we will commute the differential and expectation operator without proving the uniform integrability of the functions at hand. The reader should feel assured, however, that the condition is satisfied.

Modeling the Stock Price at Sampling Times The assumed log-normal distribution of the underlying stock prices implies that for any two moments of time $t_a < t_b$, the following relation holds:

$$S(t_b)=S(t_a)\exp\left[\left(r-\frac{1}{2}\sigma^2\right)(t_b-t_a)+\sigma Z_{a,b}\sqrt{t_b-t_a}\right].$$

Note that $Z_{a,b}$ is a value drawn from a standard normal distribution.

Consider now an Asian option with discrete price averaging, with n sampling times such that at least two sampling times occur after time 0. Now take two successive sampling times t_i and t_{i+1}, such that $0=t_0<t_i<t_{i+1}$. In simulating the price of the option at time t_{i+1}, we have the choice[2] of producing a random price starting from time 0, or for producing a random price starting from time t_i, as illustrated in the two expressions below:

$$\begin{aligned}S(t_{i+1}) &= S(0)\exp\left[\left(r-\frac{1}{2}\sigma^2\right)t_{i+1}+\sigma Z_{0,i+1}\sqrt{t_{t+1}}\right]\\ S(t_{i+1}) &= S(t_i)\exp\left[\left(r-\frac{1}{2}\sigma^2\right)(t_{i+1}-t_i)+\sigma Z_{i,i+1}\sqrt{t_{t+1}-t_i}\right].\end{aligned}$$

Are these two formulations equivalent? If not, which version should we use for modeling the evolution of the stock price? Both questions can be answered easily.

The two methods are not equivalent, in that they do not produce the same probability distribution for the prices at time t_{i+1}. A direct calculation will convince you of this. Intuitively, it is clear that the equivalence can not hold. Assume, for example, that t_i and

[2] We have other choices as well. For example, we could simulate the stock price at time t_{i+1} by starting at time t_{i-1} and "skipping" over time t_i. As the subsequent discussion makes it clear, these alternative methods would not correct. As they do not provide any additional insights, we do not consider them explicitly.

t_{i+1} are very close to each other, but far away from $t_0 = 0$. Under these conditions, it is highly likely that $S(t_{i+1})$ will be close to the price at $S(t_i)$. This condition is satisfied when we simulate $S(t_{i+1})$ starting from $S(t_i)$, but it will not necessarily hold when we start the simulation at $t_0 = 0$. The problem is that when we restart the simulation from $t_0 = 0$, we disregard the fact that the stock price has reached $S(t_i)$ and time t_i, and that this fact constrains the future evolution of the stock price (changes the distribution of the future stock prices compared to the unconstrained distribution generated by starting from t_0).

As long as we make sure that the generates prices are not independent of each other, we can still write the price for each intermediate sampling time t_i, $i = \overline{1,n}$ by using the price of the underlying stock at time t_0:

$$S(t_i) = S(0) \exp\left[\left(r - \frac{1}{2}\sigma^2\right) t_i + \sigma \sum_{l=1}^{i-1} Z_{l,l+1} \sqrt{t_{l+1} - t_l}\right].$$

Note that $Z_{l,l+1}$ are independent standard normal random variables.

The Delta of An Asian Call Using the insights gained above, we get:

$$\begin{aligned}
\Delta &= \frac{\partial V}{\partial S(0)} \\
&= \underbrace{\frac{\partial}{\partial S(0)}\left[e^{-r(T-t)}\right]}_{=0} \mathbb{E}_q\left[\max(\bar{S} - K, 0)\right] + e^{-r(T-t)} \frac{\partial}{\partial S(0)} \mathbb{E}_q\left[\max(\bar{S} - K, 0)\right] \\
&= .e^{-r(T-t)} \mathbb{E}_q\left[\frac{\partial}{\partial S(t)}\left[\max(\bar{S} - K, 0)\right]\right].
\end{aligned}$$

Focusing on the differential operator only, we can now write the following:

$$\begin{aligned}
\frac{\partial}{\partial S(0)}\left[\max(\bar{S} - K, 0)\right] &= \frac{\partial}{\partial \bar{S}}\left[\max(\bar{S} - K, 0)\right] \frac{\partial \bar{S}}{\partial S(t)} \\
&= \begin{cases} \frac{\partial \bar{S}}{\partial S(t)}, \text{ if } \bar{S} > K \\ 0, \text{ if } \bar{S} < K \end{cases} \\
&= \mathbb{I}_{\bar{S}>K} \frac{\partial \bar{S}}{\partial S(t)} \text{ a.s.}
\end{aligned}$$

Note that $\max(\bar{S} - K, 0)$ is not differentiable with respect to $\bar{S}$ at the point $\bar{S} = K$. This is not a problem, however, as the set of point of non-differentiability is of 0-measure.

The notation $\mathbb{I}_{\bar{S}>K}$ denotes the indicator function of the event $\bar{S} > K$. Computing the

value of $\frac{\partial \bar{S}}{\partial S(t)}$ is straightforward:

$$\begin{aligned}
\frac{\partial \bar{S}}{\partial S(0)} &= \frac{\partial}{\partial S(0)}\left[\frac{1}{n}\sum_{i=1}^{n} S(t_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial S(t_i)}{\partial S(0)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial S(0)} S(0)\exp\left[\left(r-\frac{1}{2}\sigma^2\right)t_{t+1}+\sigma\sum_{l=1}^{i-1} Z_{l,l+1}\sqrt{t_{l+1}-t_l}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\exp\left[\left(r-\frac{1}{2}\sigma^2\right)t_{t+1}+\sigma\sum_{l=1}^{i-1} Z_{l,l+1}\sqrt{t_{l+1}-t_l}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{S(t_i)}{S(0)} \\
&= \frac{\bar{S}}{S(0)}.
\end{aligned}$$

Putting everything together, we get that

$$\Delta = \frac{\partial V}{\partial S(0)} = e^{-r(T-t)}\frac{\mathbb{E}_q\left[\mathbb{I}_{\bar{S}>K}\bar{S}\right]}{S(0)}.$$

Ignoring non-random factors, the delta of the Asian call has been written as an expectation. By noting that a regular European call is just an Asian call with a single sampling period at the expiration date, we can also write the formula for the delta of European calls:

$$\Delta_{EU} = \frac{\partial V_{EU}}{\partial S(0)} = e^{-r(T-t)}\frac{\mathbb{E}_q\left[\mathbb{I}_{S(T)>K}S(T)\right]}{S(0)}.$$

This expression can easily be evaluated directly, through a computation similar in flavor to that undertaken when we derived the Black-Scholes formulas.

The Vega of an Asian Call We proceed in a manner analogous to the previous derivation:

$$\begin{aligned}
\upsilon &= \frac{\partial V}{\partial \sigma} \\
&= \underbrace{\frac{\partial}{\partial \sigma}\left[e^{-r(T-t)}\right]}_{=0}\mathbb{E}_q\left[\max(\bar{S}-K,0)\right] + e^{-r(T-t)}\frac{\partial}{\partial \sigma}\mathbb{E}_q\left[\max(\bar{S}-K,0)\right] \\
&= e^{-r(T-t)}\mathbb{E}_q\left[\frac{\partial}{\partial \sigma}\left[\max(\bar{S}-K,0)\right]\right].
\end{aligned}$$

Focusing on the differential operator, we can write the following:

$$\begin{aligned}\frac{\partial}{\partial\sigma}\left[\max(\bar{S}-K,0)\right] &= \frac{\partial}{\partial\bar{S}}\left[\max(\bar{S}-K,0)\right]\frac{\partial\bar{S}}{\partial\sigma} \\ &= \mathbb{I}_{\bar{S}>K}\frac{\partial\bar{S}}{\partial\sigma} \text{ a.s.}\end{aligned}$$

The derivation of the $\frac{\partial\bar{S}}{\partial\sigma}$ term is also easy:

$$\begin{aligned}\frac{\partial\bar{S}}{\partial\sigma} &= \frac{\partial}{\partial\sigma}\left[\frac{1}{n}\sum_{i=1}^{n}S(t_i)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial S(t_i)}{\partial\sigma} \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\sigma}S(0)\exp\left[\left(r-\frac{1}{2}\sigma^2\right)t_i+\sigma\sum_{l=1}^{i-1}Z_{l,l+1}\sqrt{t_{l+1}-t_l}\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(-\sigma t_i+\sum_{l=1}^{i-1}Z_{l,l+1}\sqrt{t_{l+1}-t_l}\right)S(t_i) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left\{-\sigma t_i+\frac{1}{\sigma}\left[\log\frac{S(t_i)}{S(0)}-\left(r-\frac{1}{2}\sigma^2\right)t_i\right]\right\}S(t_i) \\ &= \frac{1}{n\sigma}\sum_{i=1}^{n}\left[\log\frac{S(t_i)}{S(0)}-\left(r+\frac{1}{2}\sigma^2\right)t_i\right]S(t_i)\end{aligned}$$

Putting everything together, we write the vega of the Asian call as an expectation:

$$\upsilon=\frac{\partial V}{\partial\sigma}=\frac{e^{-r(T-t)}}{n\sigma}\mathbb{E}_q\left[\mathbb{I}_{\bar{S}>K}\sum_{i=1}^{n}\left[\log\frac{S(t_i)}{S(0)}-\left(r+\frac{1}{2}\sigma^2\right)t_i\right]S(t_i)\right].$$

Again, we can immediate derive the formula for a European call's vega:

$$\upsilon_{EU}=\frac{\partial V_{EU}}{\partial\sigma}=\frac{e^{-r(T-t)}}{\sigma}\mathbb{E}_q\left[\mathbb{I}_{S(T)>K}S(T)\left[\log\frac{S(T)}{S(0)}-\left(r+\frac{1}{2}\sigma^2\right)T\right]\right].$$

The formulas for an Asian call's delta and vega do not appear particularly simple, nor useful. Their advantages will be revealed as we examine Monte-Carlo methods below.

0.2.6 A Primer on Monte-Carlo Methods

Theorem 1 *(The Weak Law of Large Numbers) If random variables X_1, X_2, X_3, ... are independent, $\mathbb{E}[X_i]=m$, $Var[X_i]\leqslant K<\infty$, and $S_n=\sum_{i=1}^{n}X_i$, then $\frac{S_n}{n}\to m$ in $\mathbb{L}^2$ and probability.*

This, and similar theorems, provide the theoretical basis for evaluating expectations using Monte-Carlo methods. Consider the formulas we derived for Asian options:

$$\begin{aligned} V(t,T;K) &= e^{-r(T-t)}\mathbb{E}_q\left[\max(\bar{S}-K,0)\right] = e^{-r(T-t)}\mathbb{E}_q\left[\mathbb{I}_{\bar{S}>K}(\bar{S}-K)\right] \\ \frac{\partial V}{\partial S(0)} &= e^{-r(T-t)}\frac{\mathbb{E}_q\left[\mathbb{I}_{\bar{S}>K}\bar{S}\right]}{S(0)} \\ \frac{\partial V}{\partial \sigma} &= \frac{e^{-r(T-t)}}{n\sigma}\mathbb{E}_q\left[\mathbb{I}_{\bar{S}>K}\sum_{i=1}^{n}\left[\log\frac{S(t_i)}{S(0)}-\left(r+\frac{1}{2}\sigma^2\right)t_i\right]S(t_i)\right] \end{aligned}$$

Any of the quantities under the expectation operator can be identified with the random variables X_i. All we have to do is to generate "many" sample values and average them to get estimates of the respective random quantity's expectation.

But how many sample values do we need to average? A measure of how far from the true value our average is given by the standard deviation of S_n: $\sigma_{S_n} = \sqrt{Var\left[S_n\right]} = \sqrt{\frac{1}{n^2}\sum_{i=1}^{n} Var\left[X_i\right]} \leqslant \sqrt{\frac{K}{n}}$. By this estimate of the error, the convergence of Monte-Carlo methods is $O\left(n^{-\frac{1}{2}}\right)$. As we know, this is not the whole story, as asymptotic comparisons between the performance of various methods might occasionally be misleading. Put simply, the "big-O" notation hides constants, and these constants can surprise us. Still, for simple problems, this rate of convergence is too slow compared to other methods. The advantage of Monte-Carlo methods, however, is that their rate of convergence is not sensitive to many problem parameters that degrade the performance of alternative methods. Hence, Monte-Carlo methods are preferred, or at least competitive, for solving many complex problems. The valuation of Asian options is one of the simplest problems where the advantages of these methods are borne out.

It is possible to increase the rate of convergence by generating sample values of that are not entirely random. Such techniques, known as "pseudo Monte-Carlo" methods can increase the rate of convergence to (almost) $O(n^{-1})$. We do not discuss these methods further.

Of course, in practice we will often not know the variance of the quantity that we are trying to compute. We can get an unbiased estimate of the variance using the formula:

$$\widetilde{Var}\left[S_n\right] = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \frac{1}{n}S_n\right)^2.$$

These being said, let us return to the problem of the value, the delta, and the vega of an Asian call. It is clear that one can generate random sample values for $\bar{S}$. Once values for $\bar{S}$ are available, it takes only a small - and constant, per path simulated - amount of work to compute the value of expressions $\mathbb{I}_{\bar{S}>K}(\bar{S}-K)$ and $\mathbb{I}_{\bar{S}>K}\bar{S}$, which appear under the expectation operator in the formulas for the value and delta of the call. Assuming that the cost of generating $\bar{S}$ dominates the cost of these two simple expressions, it is possible to compute both the value of the call and its delta will little additional overhead.

This is in contrast with the requirement of repeating the effort of computing the option value if we had used, say, a symmetric difference method to compute delta.

Things get even better. There are no obvious simple ways to generate sample values for $\bar{S}$ without generating sample values for the stock price at all relevant times t_i, $i = \overline{1,n}$. If so, then all the elements for computing vega are generated whenever we compute either the call value, or its delta (or both). Again, if the cost of generating the sample values $S(t_i)$ dominates the cost of computing the value of the expressions under the expectation operators, then we can compute vega with minimal overhead. To compute delta and vega using finite difference approximations would require at least triple the effort needed for a simple valuation, while using the expectation form of delta and vega we can get away with little more than the effort needed for a simple valuation.

0.2.7 Generating Random Numbers

Given the distribution of a random variable, how do we sample it? In other words, how do we generate random values drawn from a given distribution? The answer to this question is surprisingly subtle and complex.

Let φ be the p.d.f., and Φ be the c.d.f. of a given distribution. From the definition of Φ, we have that

$$\Phi(z) = \text{Prob}(z^{'} \leqslant z) = \int_{-\infty}^{z} \varphi(z)dz.$$

Let u be a random variable drawn from a uniform distribution over the interval $(0,1)$, i.e. $u \tilde{} U(0,1)$.[3] Let z_u the value that satisfies the equality $\Phi(z_u) = u$; i.e. $z_u = \Phi^{-1}(u)$. It is easy to prove that z_u will have the same distribution as that specified by φ and Φ.

So, if we can generate random values drawn from a uniform distribution on the interval $(0,1)$, then we can generate variables with any distribution. But how to generate such uniformly distributed values?

The problem of generating uniformly distributed random variables is complicated by the limits of computer hardware. Let us assume for a moment that we have an algorithm that draws samples from a (mathematically) uniform random distribution, and maps them to the floating-point number in the interval $(0,1)$ closest to the generated number.[4] Let $F_{(0,1)}$ be the set of all floating point numbers that can be represented in the interval $(0,1)$. When lined up on the real axis, these values are not equidistant; specifically, values closer to the left end of the interval (0) will be closer to their neighbors, than values closer to the right end of the interval (1). This means that the size of the interval within which a point is closer to a given value f from $F_{(0,1)}$ than any other value in $F_{(0,1)}$ depends on the magnitude of f. Hence, certain values from $F_{(0,1)}$ will be chosen more frequently than others. So the fact that we can not represent the entire continuum of values in $(0,1)$ impacts the uniform

[3] Note that we did not include the ends of the interval, i.e. values 0 and 1. Can you see why?

[4] Of course, if such an algorithm existed, it could not work this way. This is because the algorithm would have to use the very same floating point representation whose limits we are now discussing. The algorithm could generate its values, however, as if this mapping would truly occur after a corresponding value has been drawn from the underlying truly uniform distribution.

distribution in a subtle way. The inherent discretization introduced by the floating point representation could negatively influence the quality of the distributions produced using $\Phi^{-1}(u)$. Could it be that the non-uniform random distributions that we generate are in some sense worse if we start from our "uniform" distribution that if we had generated them directly using some alternate method? These are important questions, but we do not elaborate on them.

We can, in principle, generate random values in $(0, 1)$ that are uniform in the sense that all values that can be produced are equally likely to be chosen. This can be done, for example, by dividing the interval $(0, 1)$ into a number of equal-length subintervals, and choosing the midpoint of each such interval as the representative for the interval. In the case of the floating point representation with a mantissa of N bits, for example, we can set the exponent to 0, and then create an algorithm that generates all possible bit configurations for the mantissa.[5]

Except for scaling by 2^N, the problem of generating the uniform distribution is reduced to selecting any random integer in the range 1 to $2^N - 1$ with equal probability. Many random number generators use similar ideas.

We have reduced the problem of generating a uniform random distribution on $(0, 1)$ to the problem of generating integers in a given range. The latter problem has been studied extensively.

Since our computers are deterministic, true randomness is hard to come by. Assume that we have a source that produces one random bit b at a time, such that Prob($b = 1$) =Prob($b = 0$) = 0.5. By generating a sequence of N such random bits, we can generate all numbers in the range from 1 to $2^N - 1$ (but we have to throw away the configurations for 0).

So how do we generate a truly random bit b with the property that Prob($b = 0$) = 0.5? It turns out that we can reduce this problem to the problem of generating a random bit $\mathbf{b}$ with the property that Prob($\mathbf{b}$) = p, where p is fixed, but unknown. Indeed, assume that we generate two biased random bits $\mathbf{b}_1$, and $\mathbf{b}_2$. As long as $\mathbf{b}_1 = \mathbf{b}_2$, we keep generating a new pair of bits. If $\mathbf{b}_1 \neq \mathbf{b}_2$, then we set b to the value of $\mathbf{b}_1$. Show that this method generates an unbiased bit from a biased bit!

Generating a truly random - but perhaps biased - bit is still not feasible in a deterministic computer. This feat is typically achieved using special hardware. Hardware that produces random bits by sampling the electronic noise produced by special-purpose circuit elements is commercially available, but often expensive.

It turns out that for many applications true randomness is neither needed, nor particularly useful. Indeed, it is often sufficient to generate pseudo-random sequences of integers. Such sequences are otherwise perfectly deterministic, but they appear as if they were random in that they pass statistical tests that a truly random sequence would be expected to pass. Pseudo-random sequences are often "seeded" with values, which - while

[5]Floating point representations normalize the mantissa, so that the first digit in base 2 or 16 is non-zero. Normalization might force us to use a non-zero exponent for some values that we generate, but this is a detail that we can easily account for.

deterministic themselves - are not easily predictable or reproducible (unless the user reuses such a value directly). Good seeds are provided by the total time, measured in miliseconds or microseconds, elapsed from a reference date, by the time of continuous computer uptime, or by the number of microprocessor instructions executed since the last reboot. Saved seeds allow for the repetition of the entire pseudo-random sequence. This is a great advantage when testing and debugging programs, for example.

As the brief discussion above shows, generating random or pseudo-random number sequences is not trivial. Without expanding further, we conclude by paraphrasing two pieces of advice from Donald Knuth's "Art of Computer Programming:"

a) Do not trust your random number generator; test it. Indeed, it is not uncommon for random number generators, especially for those in general-purpose packages, to be flawed. This problem has been mitigated by the development of high-quality mathematical software in the past few years, but "surprises" in this area are still likely to be abundant.

b) Do not choose your random number generator randomly!