# 11. Beyond Vanilla Options

*Other option payoffs*

As we have seen, vanilla options provide exposure to a variety of equity and FX risks—either for speculation or hedging—and do so with a relatively small outlay of capital. But while the common European and American versions of these options are certainly useful, there are a host of other option styles that have been devised to target certain more complex risks held by a variety of market participants, or to enable extremely tailored exposures that are usually even less expensive than vanilla options.

Sometimes these alternate payoffs may be assigned the label of "exotic options," but in the course of time, it often happens some instruments once considered exotic become common in particular markets due to their utility. Alternatively, sometimes a particular payoff may enjoy a brief vogue and fall out of favor.

From a practitioner's standpoint, of course one of the attractions of these alternative payoffs is that they are…well, fun, since they offer new problems and challenges. Some may have closed-form formulas, but at times the assumptions required to arrive at these formulas are in contradiction with the main purpose of the option's existence. Too, it is not difficult to make small modifications to a payoff with a closed-form formula and transform it into one that does not—witness the humble American put option, for example. As a result, while closed-form valuation is always desirable, numerical methods tend to be more robust and more flexible tools to bring to bear on such problems.

What follows are just three examples of payoffs that were once considered "exotic," along with illustrations of methods that may be used to price them. Two of these no longer truly deserve the label, however, and are now rather common. The third is an example of an option that has (thus far) never really "caught on" due mainly to its expense, but it illustrates a useful numerical technique that can be applied to a very wide variety of problems.

*Barrier options*

A barrier option is a variation on the vanilla European option payoff with an additional provision: The option is specified with a barrier level $H$ that either must be crossed or must not be crossed in order for the vanilla option to attain its usual value. If $H$ is a level that must be crossed, it is referred to as a "knock-in" barrier; if it must not be crossed, it is referred to as a "knock-out" barrier.

To make it explicit what it means for the barrier to be crossed, typically the options are named in a way that indicates the spot price's relationship to the barrier at the time the deal was entered into. So, for example, a "down-and-in put" is a European put option that comes into existence only if the barrier, which was initially below the spot price, is crossed. If the barrier is never crossed—i.e., the option never knocks in—then the option expires worthless.

This gives rise to a number of possible position combinations, some of which make no practical sense as deals:

| calls | up and… | H > S0 |
|---|---|---|
| | K < H | K > H |
| in | valid | call w/ strike K |
| out | valid | worthless |

| puts | up and… | H > S0 |
|---|---|---|
| | K < H | K > H |
| in | valid | valid |
| out | valid | valid |

| calls | down and… | H < S0 |
|---|---|---|
| | K < H | K > H |
| in | valid | valid |
| out | valid | valid |

| puts | down and… | H < S0 |
|---|---|---|
| | K < H | K > H |
| in | put w/ strike K | valid |
| out | worthless | valid |

So of the 16 possible combinations up / down, in / out, put / call, 4 are degenerate; the remaining 12 deals are all valid. One of the things we see particularly clearly in the degenerate cases, however, is that for each such case we have a companion case in which the value of the “barrier” option that is simply the value of a vanilla option. This fact suggests a property of barrier options is called in-out parity.

The reason behind this relation is fairly obvious: Consider a portfolio consisting of, for example, an up-and-out call option with strike *K* and barrier level *H*, and an up-and-in call option with the same strike and barrier level. This portfolio will under all circumstances (whether or not the barrier is crossed) pay the same as a vanilla call option. Thus, if we wish to know the price of the up-and-in call, for example, it suffices to know the price of the up-and-out call with the same strike and barrier level and the price of the vanilla call at the same strike.

Under Black-Scholes assumptions, these options actually have closed-form formulas. For example, the following is the valuation formula in this context for a down-and-out call:

$$C_{do} = S_0 e^{-qT} N(x_1) - K e^{-rT} N\left(x_1 - \sigma\sqrt{T}\right) - S_0 e^{-qT}\left(\frac{H}{S_0}\right)^{2\lambda} N(y_1) + K e^{-rT}\left(\frac{H}{S_0}\right)^{2\lambda-2} N\left(y_1 - \sigma\sqrt{T}\right)$$

$$\lambda = \frac{r - q + \frac{\sigma^2}{2}}{\sigma^2}$$

$$x_1 = \frac{\ln\left(\frac{S_0}{H}\right)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$$

$$y_1 = \frac{\ln\left(\frac{H}{S_0}\right)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$$

This is one of 6 such formulas, corresponding to the 12 cases we saw above where the specification of the deal is valid; to obtain the other 6, we use in-out parity.

While it is no doubt an achievement to have obtained these formulas, their practical use is, unfortunately, limited. There are a large number of slight variations on this instrument traded in the market: Some may have partial barriers, which apply only over part of the time until expiry; some may have stepped barriers, in which the barrier level changes over time; still others may have soft barriers, where the barrier is defined as a region where the percent of the region not “knocked out” during the term of the option dictates the percent of the vanilla payoff received.

More importantly, the Black-Scholes assumption of the same volatility at all strikes emphatically does not hold in the FX market, where these deals are most common. This difference becomes particularly noticeable in the context of barrier options, whose values are often quite small to begin with, magnifying the effect of volatility assumptions. In short, the only real practical way of pricing these on a desk that trades them is a reliable and flexible numerical method.

*Trinomial trees*

As we saw earlier, binomial trees are an extremely efficient numerical method for pricing certain deals, but it has some undesirable properties. We have seen that the tendency for convergence to be oscillatory can be mitigated, but even so the fact that these methods do not cover the same set of points for the underlying at each time step can make their behavior quite strange for instruments such as barrier options.

A trinomial tree is a nice compromise between the approachability of a lattice method and the flexibility of a full-fledged finite difference method, to which it is closely related. In a trinomial tree, the underlying can transition to any of three states: up, down, or unchanged. This arrangement means that, aside from the tree’s natural expansion through time, the range of spot prices covered at each time step is the same. Even more usefully, the fact that there are three probabilities involved means that there is in essence a free parameter in the specification of the tree. This can be used to great effect for time-varying parameters or even more complicated arrangements.

For our purposes, we consider a common parameterization used to represent the Black-Scholes assumptions…

$$u = e^{\sigma\sqrt{3\Delta t}}$$

$$d = \frac{1}{u}$$

$$p_u = \sqrt{\frac{\Delta t}{12\sigma^2}}\left(r - q - \frac{\sigma^2}{2}\right) + \frac{1}{6}$$

$$p_m = \frac{2}{3}$$

$$p_d = -\sqrt{\frac{\Delta t}{12\sigma^2}}\left(r - q - \frac{\sigma^2}{2}\right) + \frac{1}{6}$$

…where $p_m$ is the probability of the spot remaining unchanged, and the others as expected the probabilities of the up and down transitions.

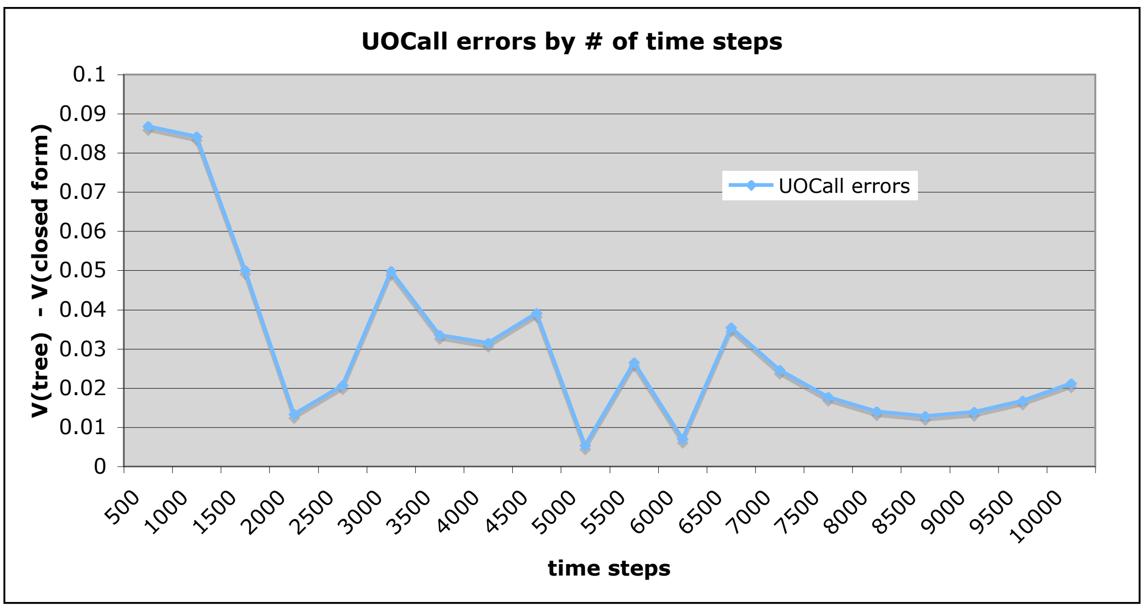
Otherwise, the operation of such a tree is quite similar to that of a binomial tree. One immediate benefit of the trinomial arrangement is that vanilla payoffs no longer exhibit oscillatory convergence. However, the increase in the number of nodes means that these trees entail significantly more computational expense than binomial trees do.

*Application of trinomial trees to a barrier option*

Suppose we wish to price an up-and-out call when the spot price is 100, with strike level 100, barrier level 110, and three-month expiry. We assume a constant interest rate of 50 basis points, a dividend rate of 20 basis points, and a volatility of 15%.

Note that knockout options can be priced in the most straightforward way using a lattice method: The value of the option at every point at or beyond the barrier is zero by construction. In order to price a knock-in option, the clearest method to use is to price a vanilla and a knockout and report the difference as the price of the knock-in.

However, a naïve approach to pricing even the simpler knockout option can lead to problems, as illustrated below:




The value of the option is only 1.03937, so these errors are hardly negligible. But why do they arise?

For any lattice method, the valuation experiences discretization errors—that is, errors that arise from using a discrete set of points to approximate data that is, in concept, continuous. We have already seen these kinds of errors in binomial trees and the patterns to which they lead. But while these errors surely exist in this case, they are not responsible for the size of what we see here, or the unmistakable pattern that our tree is returning a value that is too large in every case.

This is sometimes called the "specification error" of pricing in connection with barrier options. Consider the layer of spots near the barrier level *H*: Immediately below is a level that, at payoff, has the maximum call value. The closer this level lies to the barrier, the higher this value is. As we move backwards through the tree, all of the up transitions from this level have value zero, but the unchanged and down transitions have positive value as long as we have a fine enough discretization. In other words, if we call $H_-$ the level immediately below the barrier, this model returns precisely the same price for all barrier options with barrier level in the interval ($H_-$, $uH_-$).

Moreover, the closer $H_-$ lies to the actual barrier without actually touching it, the more extreme this error looks likely to be. At times when the spot price is extremely close to the barrier, the value of the option should be near zero; instead, it is assigning artificially high prices to these options. Thus, the result of the naïve implementation is a price that is too high. Furthermore, since the only free parameter in the parameterization detailed above is the number of time steps, it is not easily possible to find a number of time steps near our wanted level that will minimize this error.

Fortunately, the additional free parameter in a trinomial tree can be used to overcome this shortcoming if we use an alternate parameterization. The one below is by no means the only one available, but it will work: This takes the size of the up step *u* as a free parameter:

$$d = \frac{1}{u}$$

$$p_u = \frac{1}{u+d-2}\left(e^{(r-q)\Delta t} + c(d-1) - 1\right)$$

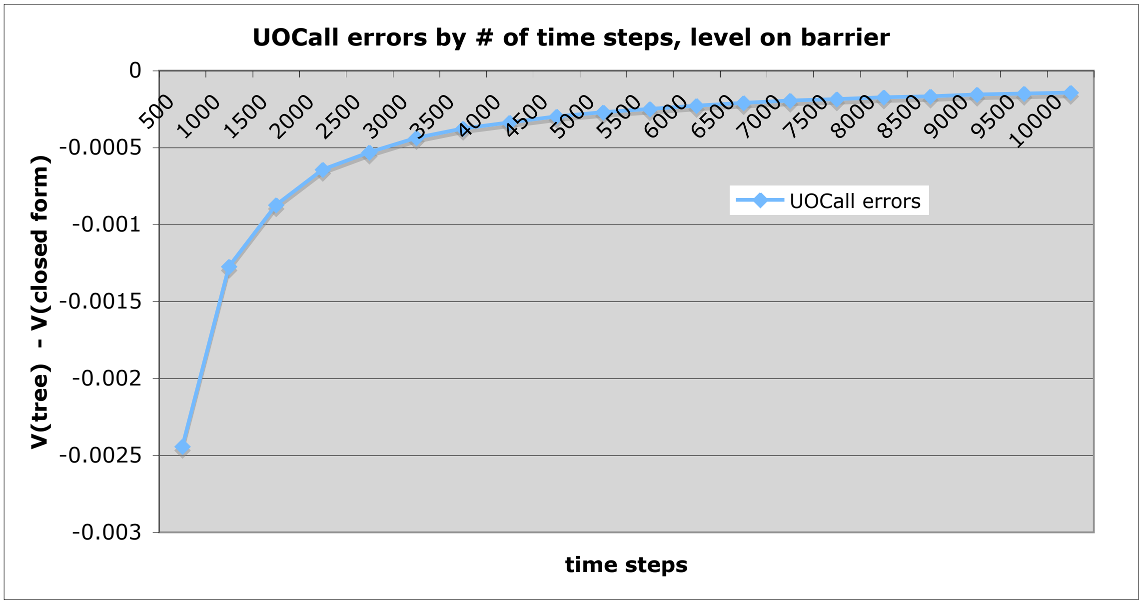
$$p_m = 1 - p_u - p_d$$

$$p_d = p_u - c$$

$$c = \frac{1}{\ln u}\left(r - q - \frac{\sigma^2}{2}\right)\Delta t$$

Clearly, this does not converge for all choices of *u* and $\Delta t$, but no trinomial tree scheme is convergent for all choices of input.

With *u* a free parameter, we can now choose a value that, if we wish, is very close to the value recommended by our parameterization above and then adjust it slightly to ensure that one level of our tree will lie exactly on the barrier. We have done this below, using a markedly smaller *u* than recommended by our initial parameterization, since the bulk of the values we are interested in lie in a relatively narrow band around the forward:




Clearly the convergence and the results, even for a relatively low number of time steps, are far more satisfactory when the problem is approached this way. Also, we see in this arrangement that discretization error causes us to consistently undershoot the theoretical price, which is not entirely surprising. Further refinements of the method used above could produce even better performance with relatively little additional effort.

Generally speaking, the value of such options to those who hold them is their cheapness relative to vanilla options of similar type. By in effect selling back a range of outcomes, quite specific exposures can be manufactured. The downside of this from a pricing—and from a hedging—standpoint is that the instrument must be handled quite carefully. While finite difference methods are quite powerful in this regard, it may be more difficult to accommodate the variety of variations traded using this approach than a more straightforward lattice method, although as we have seen it must be used carefully.

*Spread options*

A spread option is an option on the difference in price between two reference assets at expiry. These occur in a variety of markets but are particularly important in interest rates and commodities. In the context of interest rates, these can be constructed to provide inexpensive exposure to the slope of the yield curve; in commodities, they are a natural way of managing the roll risk encountered when a position in one month's contract is closed and the equivalent position in the following month's contract is entered into.

If the assets have prices $S_1$ and $S_2$, then the payoff of a spread call at maturity is $Max\ (S_{1,T} - S_{2,T} - K)^+$. Under the assumption that the prices themselves are normal, this would be a straightforward payoff to price. However, under the usually more relevant assumption that these terminal prices are each lognormally distributed, we encounter the difficulty that the difference of lognormally distributed random variables has no conveniently specified distribution.

Furthermore, as suggested by the illustrations above for the uses of these contracts, these assets' returns are likely to be fairly strongly correlated, which is a feature we will also need to accommodate in any pricing method we employ. In this case, we can make use of the lognormal assumption to our advantage. If the risk-neutral evolution of the first asset is given by…

$$\frac{dS_1}{S_1} = (r - q_1)dt + \sigma_1 dW_1$$

…for some Wiener process, then the evolution of a second asset with constant correlation ρ to this one can be written as…

$$\frac{dS_2}{S_2} = (r - q_2)dt + \rho\sigma_2 dW_1 + \sqrt{1-\rho^2}\sigma_2 dW_2$$

…where $W_2$ is another Wiener process independent of the first. Furthermore, since there is no need to simulate the entire path of either asset for this European option, we can directly specify the terminal distributions…

$$S_{1,T} = S_{1,0} \exp\left\{\left(r - q_1 - \frac{\sigma_1^2}{2}\right)T + \sigma_1\sqrt{T}Z_1\right\}$$

$$S_{2,T} = S_{2,0} \exp\left\{\left(r - q_2 - \frac{\sigma_2^2}{2}\right)T + \sigma_2\sqrt{T}\left(\rho Z_1 + \sqrt{1-\rho^2}Z_2\right)\right\}$$

…for independent standard normal random variables $Z_1$ and $Z_2$. This suggests one quite straightforward way of pricing the option: Monte Carlo.

*Monte Carlo pricing of a spread option*

For the particular examples here, we choose assets and a contract with the following properties:

$$S_{1,0} = S_{2,0} = 100$$

$$r = 0.02$$

$$q_1 = q_2 = 0$$

$$K = 2$$

$$T = 0.5$$

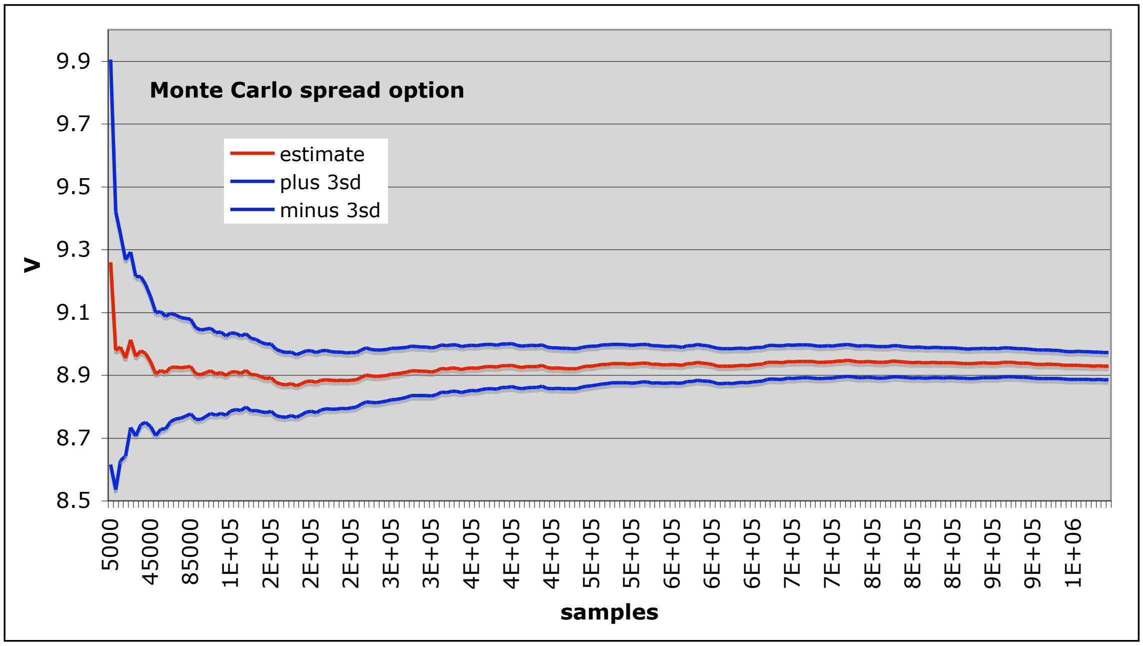
$$\sigma_1 = 0.3$$

$$\sigma_2 = 0.25$$

$$\rho = 0.2$$

For each random trial, we draw two independent standard normals as described above (this can be done using an inverse CDF method, or the somewhat faster Box-Muller method that returns independent standard normals in pairs naturally), evaluate the payoff and discount. This result is a random variable $X$ whose expected value is the quantity of interest $V$—the present value of the spread option. To estimate this, we take the sample mean $X_{avg,N}$ over $N$ trials, and by the central limit theorem we know that this statistic is for large $N$ distributed with mean $V$ and standard deviation stdev($X$) / sqrt($N$). Although we do not directly know the standard deviation of $X$ *a priori*, we can use the sample standard deviation as an estimator and define an interval—three standard deviations is common—within which we are quite confident the actual value of the option lies. If our goal in pricing is to know the price within a certain degree of precision, this can be used to define a stopping criterion for the Monte Carlo iteration.

The below illustrates how rapidly the interval described above shrinks for a particular attempt to price the spread option:




As is usual with Monte Carlo, we reach a decent estimate fairly quickly. And, as is also usual with Monte Carlo, significant improvement on that estimate requires many, many further iterations. While the central limit theorem guarantees us we will reach the correct value in time, it also tells us we can expect to need to perform 100 times as many iterations as we have already performed in order to add a decimal point of accuracy to our estimate of that value.

*Use of a control variate*

There are many techniques available for improving the convergence speed of Monte Carlo methods. A simple one is the use of a control variate. This is a quantity $Y$ with known expected value $\mu_Y$ whose value in each trial is related to the quantity of interest $X$. We then perform our Monte Carlo calculations on the quantity, using some multiplier $k$:

$$X_{CV} = X - k\left(Y - \mu_Y\right)$$

The additional term can help to control for excessive or idiosyncratic variation in the Monte Carlo draws used. Clearly, this has no effect on the expected value of the test statistic…

$$E\left[X_{CV}\right] = E\left[X\right] - E\left[k\left(Y - \mu_Y\right)\right] = \mu_X$$

…but does have an effect on its variance:

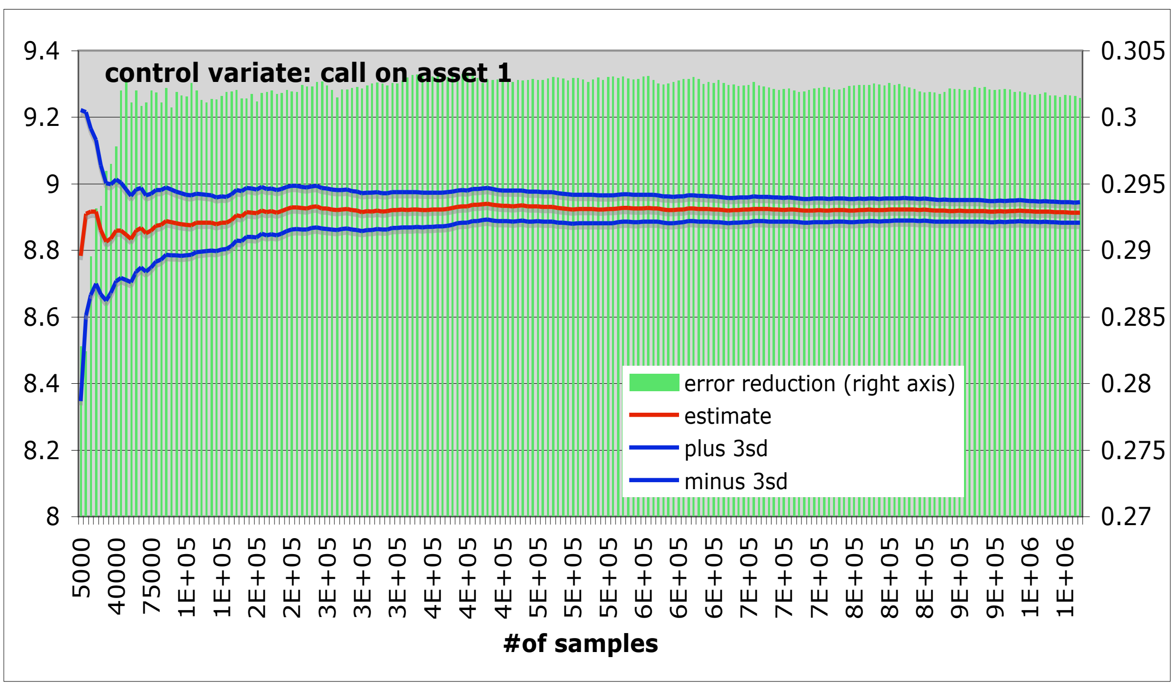
$$Var\left[X_{CV}\right] = Var\left[X\right] + Var\left[k\left(Y-\mu_Y\right)\right] - 2Cov\left[X, k\left(Y-\mu_Y\right)\right]$$
$$Var\left[X_{CV}\right] = \sigma_X^2 + k^2\sigma_Y^2 - 2k\rho_{X,Y}\sigma_X\sigma_Y$$

We then attempt to choose $k$ to minimize the variance of the new simulation variable:

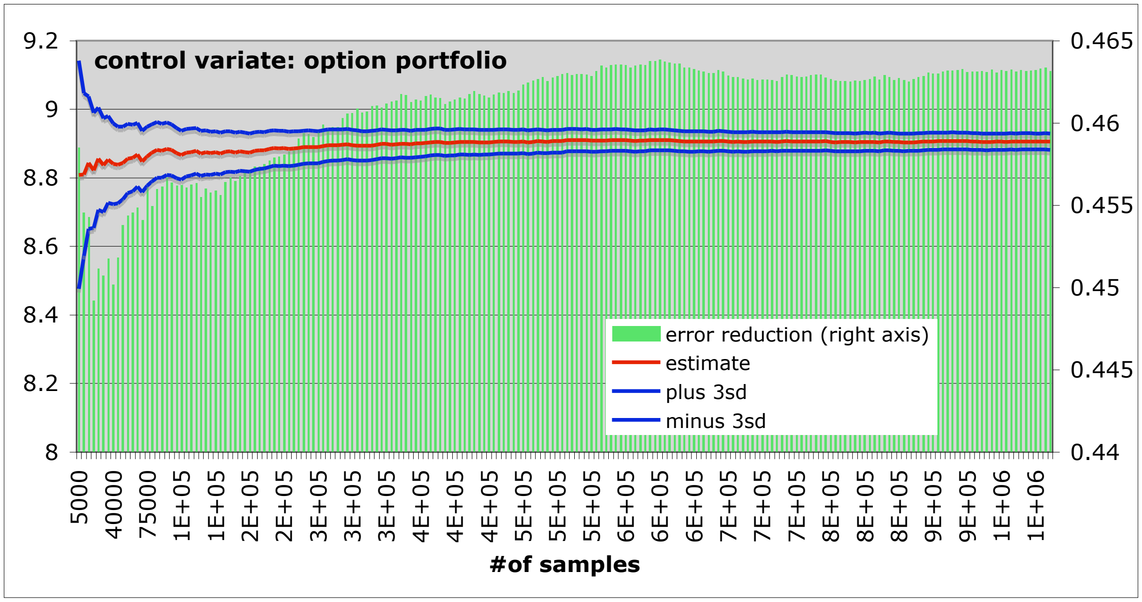
$$\frac{d}{dk}Var\left[X_{CV}\right] = 2k\sigma_Y^2 - 2\rho_{X,Y}\sigma_X\sigma_Y$$
$$2\hat{k}\sigma_Y^2 - 2\rho_{X,Y}\sigma_X\sigma_Y = 0$$
$$\hat{k} = \rho_{X,Y}\frac{\sigma_X}{\sigma_Y}$$

This optimal $k$ the slope of an ordinary least squares regression between the value of $X$ and the value of $Y$. While the values on the right-hand side are not known at the beginning of the simulation, if a reasonable initial choice is made then sample statistics can be used during the simulation to update the value of $k$ using the above formula, improving the degree of variance reduction as the simulation proceeds.

Even without adjustment of $k$, a good choice of instrument can help. For example, suppose we fix $k = 1$ and use as our variate the price of a call on asset 1, struck at $K$ plus the forward price of asset 2:




This is indeed helpful, reducing the error of our estimate by 30%. A final choice to consider is a portfolio of options: Here, we take a long call on asset 1 as above plus a long put on asset 2 struck at the forward price of asset 1 minus the strike $K$:




This is even more helpful, cutting the standard error of the estimate nearly in half. Using this method and 100 million samples, we estimate the value of the option to be 8.9119, plus or minus 0.00233.

*A quasi-analytic method*

The control variates we chose above suggest another possible choice of method for pricing this option: In terms of the second asset, this instrument can be considered a put with the random strike $S_{1,T} - K$. Thus, conditional on the value of $S_{1,T}$, we should be able to use a closed form to determine the value of this option. It should, then, be possible to discretize the possible outcomes in asset 1, use analytic methods to price in each of these cases, and determine the value of the option by summing these, weighted by a probability of each $S_{1,T}$ outcome.

In order to do this, we must first discretize the outcomes in the first asset. There are multiple ways to go about this: Here, we will discretize the outcomes of the standard normal variable $Z_1$ that drives the first asset's return in such a way that we make sure to match the first two moments of the distribution. (We will actually match three, since our choice of discretization will be symmetric, and thus also match the zero skewness of the distribution.)

If we wish to have $N$ points in our discretization, then we will choose $N - 1$ cutoff points that segment the real line into bins of equal probability. Thus, our cutoff points are…

$$c_i = N^{-1}\left(\frac{i}{N}\right), i \in \{1,2,\ldots,N-1\}$$

…where the function $N^{-1}$ is the inverse CDF of the normal distribution. This selection will be symmetric around the origin.

Within each interval, we will select a single node that represents all of the probability in that interval. A simple choice is the midpoint…

$$z_{1,i} = \frac{c_{i-1} + c_i}{2}, i \in \{2,3,...,N-1\}$$

…but the center of mass is also an appealing choice, and easy to calculate:

$$z_{1,i} = \frac{\frac{1}{\sqrt{2\pi}} \int_{c_{i-1}}^{c_i} x e^{-\frac{1}{2}x^2} dx}{\frac{1}{\sqrt{2\pi}} \int_{c_{i-1}}^{c_i} e^{-\frac{1}{2}x^2} dx} = \frac{N}{\sqrt{2\pi}} \int_{c_{i-1}}^{c_i} x e^{-\frac{1}{2}x^2} dx$$

$$u = -\frac{1}{2}x^2$$

$$z_{1,i} = \frac{N}{\sqrt{2\pi}} \int_{-\frac{1}{2}c_{i-1}^2}^{-\frac{1}{2}c_i^2} -e^u du = \frac{N}{\sqrt{2\pi}} \left( e^{-\frac{1}{2}c_{i-1}^2} - e^{-\frac{1}{2}c_i^2} \right), i \in \{2,3,...,N-1\}$$

This collection is symmetric around the origin, with the central node located at the origin for $N$ odd. Since we will require that the leftmost node $z_{1,1} = -z_{1,N}$, the entire discretization is symmetric and thus guaranteed to match the first moment of the standard normal distribution.

To match the second moment, we require that…

$$\frac{1}{N} \sum_{i=1}^{N} z_{1,i}^2 = 1$$

$$2z_{1,1}^2 + 2 \sum_{i>1, z_{1,i}<0} z_{1,i}^2 = N$$

$$z_{1,1} = -\sqrt{\frac{N}{2} - \sum_{i>1, z_{1,i}<0} z_{1,i}^2}$$

…and now have a discretization of the standard normal distribution where each point is assigned equal probability.

For a given choice of $z_{1,i}$, we must determine the value of a put option with strike…

$$K_i = S_{1,0} \exp\left\{ \left( r - q_1 - \frac{\sigma_1^2}{2} \right) T + \sigma_1 \sqrt{T} z_{1,i} \right\} - K$$

…where the underlying has the distribution:

$$S_{2,T,i} = S_{2,0} \exp\left\{ \left( r - q_2 - \frac{\sigma_2^2}{2} \right) T + \sigma_2 \rho \sqrt{T} z_{1,i} + \sigma_2 \sqrt{1-\rho^2} \sqrt{T} Z_2 \right\}$$

This is actually still lognormal, but it has a smaller volatility than $S_2$ alone due to the fact that some of its variation is "already determined" by the value $z_{1,i}$, and we also have an additional term due to this determination. This in effect changes the drift of the asset, and one way to approach the problem is to try making the drift adjustment in the term $q_2$:

$$\left(r - q_2 - \frac{\sigma_2^2}{2}\right)T + \sigma_2 \rho \sqrt{T} z_{1,i} =$$

$$rT - q_2 T - \frac{\sigma_2^2}{2}T + \frac{\sigma_2^2 \rho^2}{2}T - \frac{\sigma_2^2 \rho^2}{2}T + \frac{\sigma_2 \rho z_{1,i}}{\sqrt{T}}T =$$

$$rT - \left(q_2 + \frac{\sigma_2^2 \rho^2}{2} - \frac{\sigma_2 \rho z_{1,i}}{\sqrt{T}}\right)T - \frac{\sigma_2^2\left(1-\rho^2\right)}{2}T =$$

$$\left(r - q_{2,i} - \frac{\sigma_2^2\left(1-\rho^2\right)}{2}\right)T$$

Thus, we may value the put option for a particular choice of $z_{1,i}$ by the familiar Black-Scholes formula…

$$V_i = K_i e^{-rT} N\left(-d_2\right) - S_{2,0} e^{-q_{2,i}T} N\left(-d_1\right)$$

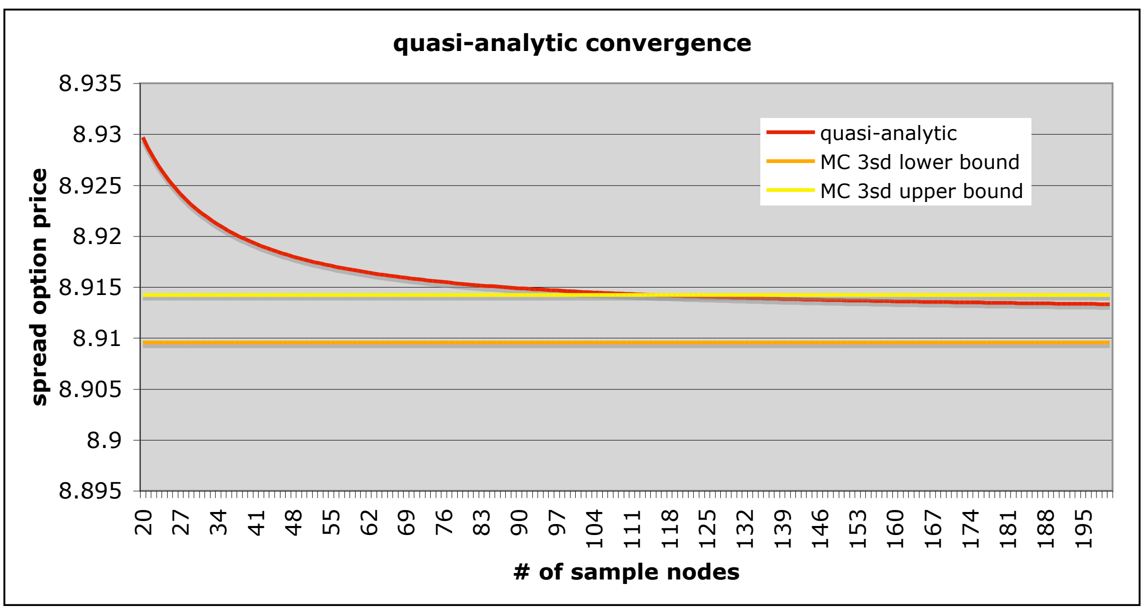
…where:

$$K_i = S_{1,0} \exp\left\{\left(r - q_1 - \frac{\sigma_1^2}{2}\right)T + \sigma_1 \sqrt{T} z_{1,i}\right\} - K$$

$$q_{2,i} = q_2 + \frac{\sigma_2^2 \rho^2}{2} - \frac{\sigma_2 \rho z_{1,i}}{\sqrt{T}}$$

$$\sigma_{BS} = \sigma_2 \sqrt{1-\rho^2}$$

Putting it all together, we achieve the following for our example with increasing choices of $N$, the number of points in the discretization of asset 1:




As we can see, in this case the discretization error creates systematically higher prices than the value found by Monte Carlo, suggesting that the correction for the tails could likely be tuned to achieve even better convergence. Still, the quasi-analytic method lands within the standard error of a 100 million-sample Monte Carlo run simply by pricing 115 options, which is instantaneous by comparison.

*Lookback options*

A lookback option is a payoff on the maximum or minimum price attained by the asset over the term of the deal. While many different variations on the payoff can be imagined, two major ones pay on the basis either of a fixed strike—i.e., $S_{max} - K$ for a call, $K - S_{min}$ for a put—or a floating strike—$S_T - S_{min}$ for a call, $S_{max} - S_T$ for a put. Such options tend to be quite expensive, which is why they are seldom traded.

Still, they provide a simple illustration of a technique that is useful for adapting lattice methods to some path-dependent payoffs. And while all of these options can be valued in closed form under the Black-Scholes assumptions, the reality that volatility is not constant to all strikes means that some more elaborate model of the underlying is required in order to value payoffs in a way that is consistent with the market. Numerical methods are in many cases the only practical way to achieve such prices.

Previously, we have used the lattice as a method of tracking only one variable of interest: the price of the underlying. At a given node, it does not matter what sort of transition—or history of transitions—caused the spot price to arrive at that node. In this case, however, the option is path-dependent, and so we need more information than merely the spot price: Namely, we must know the running maximum in order to completely define the state.

Lattice methods can be used for options of this type as well, but we must increase the dimension of the method in order to track this additional item of state information. For each time $t$ and spot level $S_t$, we will have a vector of running maxima $M$. For a trinomial tree, this means that the up, down, and mid transitions will be to nodes defined by three state variables: $(t, S_t, M)$. The rule for changes in $M$ is fairly straightforward: Each transition will be to a node in the next time step with the same running maximum $M$ unless the value $S_t$ at that node is greater than $M$, in which case we will look at the node with that running maximum. This means that the running maximum will only change on an up transition from a state in which the spot price equals the running maximum.

The result is a tree that actually consists of a number of "layers." For a lookback option with a large number of time steps, this approach can be computationally very expensive even if taken in a smart way. However, there are a variety of path-dependent options whose state can be represented in relatively few layers, and for these a lattice method with an augmented state space can be an efficient and straightforward way of pricing the option.