CS522 - Option Pricing

0.1 Putting Everything Together

The payoff of options depends in a straightforward manner on the price of the underlying stock at the time of exercise. It is thus reasonable to believe that being able to analyze the various properties of options is contingent upon the ability to model the evolution of the underlying stock's price.

We have introduced a simple model for the evolution of the stock prices. In essence, we fix an interval $[0, T]$ in which we want to model the stock price, then we subdivide it into many small subintervals Δ. We defined the return on each such subinterval as the natural logarithm of the ratio between the stock price at the end of the subinterval, and the price at its beginning. The returns on each (non-overlapping) subintervals are assumed to be independent and identically distributed. More, we assume that the return has two crucial properties:

$$\begin{aligned} E[r(i\Delta)] &= \mu\Delta \\ Var[r(i\Delta)] &= \sigma^2\Delta \end{aligned}$$

Thus the average return is proportional to the length of the interval, similarly to the variance. The simple behavior assumed on subintervals makes the return over the entire interval $[0, T]$ to be normally distributed,[1] with the expectation and variance proportional to T:[2]

$$\begin{aligned} E[r_T] &= \mu\Delta \\ Var[r_T] &= \sigma^2\Delta \end{aligned}$$

While the stock price return is normally distributed, the stock price itself is log-normally distributed. This is not a major complication.

Now we have a theoretical model for the distribution of stock prices: Can we model this distribution? Yes.

We start by modeling a simple normal distribution. This can be achieved using a simple recombining lattice, which we call the binomial model. The lattice is built up by combining one-step binomial lattices. These have a "before" (or "initial") state, and two final states: "up" and "down." The transition from the initial state into one of the final states is random; one moves from the initial state to the "up" state with a probability p.

We have illustrated empirically[3] the well known fact that a recombining lattice can

[1] Here - and below - we are focusing on the main ideas. For brevity and clarity, we will often omit details and qualifications. In this case, for example, we do not mention in the main text that the convergence to normality only holds in the limit, when $\Delta \to \infty$.

[2] We have not been entirely consistent with our notation in earlier handouts. Z_T, the return over the entire interval $[0, T]$ has later been denoted by r_T.

[3] Look up the graphs comparing the normal distribution to the binomial distribution for different values of n (the number of steps in our lattice, or the total number of events), and p!

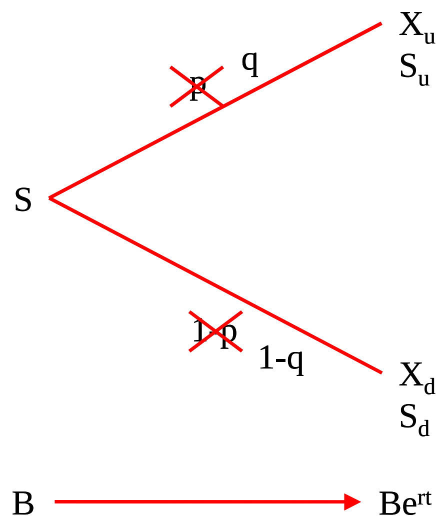

Figure 1: One-period binomial model.

be used to approximate a normal distribution. The finer the lattice (the more steps are there between the initial state and the final states), the better the approximation.

We then focused on studying the one-step binomial model. We assumed that we can invest in two instruments, a riskless money market account (or bond), whose return is fixed and known in the initial state, and a risky stock. The price of the risky stock is known in the initial state, but its final state depends on whether we end up in the "up" or "down" state. The one-period binomial model is illustrated in figure 1.

We concluded that under the "no arbitrage" assumptions, a suitably chosen portfolio of stocks and the money market can reproduce any payoff we could define in the final state. More, we were able to infer the time-0 value of the final state payoff, which is the time-0 value of the replicating portfolio. Most interestingly, we concluded that the value of the payoff does not depend on the probability p. In fact the value can be expressed as the time-t expectation of the payoff X **discounted** at the constant return of the money market account, where the expectation is based on the equivalent martingale probability q.

$$\begin{aligned} q &= \frac{Se^{rt} - S_d}{S_u - S_d} = \frac{e^{rt} - \frac{S_d}{S}}{\frac{S_u}{S} - \frac{S_d}{S}} \\ V &= e^{-rt}\mathbb{E}_q[X] = \mathbb{E}_q[e^{-rt}X] \end{aligned}$$

Let us now turn away for a second from the one-period model and focus on the multi-period lattice. We know in principle that such a lattice can be used to generate normal distributions, but we do not yet know how to generate a given distribution with parameters μ and σ.

First, we must understand how we can relate prices to the nodes of a recombining lattice. It is easy to see that if each "up" or "down" transition corresponds to multiplying[4] the price by a constant factor U or D, respectively, then paths that correspond to the same number of "up" and "down" transitions will generate the same price. This is fortunate, since the recombining lattice has this property by design.

We now know how to generate a recombining lattice, so we focus on the issue of generating the right distribution. Given parameters μ and σ, can we choose U, D, and p so that the lattice approximates a log-normal distribution of the final prices? Using Δ to denote the length of the subintervals that correspond to one step in the lattice, let us make the following choices:

$$\begin{aligned} U &= \exp(\mu\Delta + \sigma\sqrt{\Delta}) \\ D &= \exp(\mu\Delta - \sigma\sqrt{\Delta}) \\ p &= \frac{1}{2} \end{aligned}$$

With these choices, we have shown that in the limit (when $\Delta \to 0$, or equivalently, when $n \to \infty$[5]) the resulting (i.e. final) price distribution is indeed log-normally distributed with parameters μ and σ. This, of course, also means that the total stock price returns over the interval $[0, T]$ are normally distributed with parameters μ and σ.

It is important to note that the choice of U, D, and p is not unique. Indeed, one can find other parameter combinations in the literature. The fact that the choice of values is not unique is not a problem, however, as long as in the limit we get the same distribution of prices and returns.

Given μ and σ we can build a lattice that approximates a log-normal price distribution defined by these two parameters. The problem is, the true price distribution is not relevant for pricing. You can see this by returning to the one-period binomial model. The mean and variance of the "true" distribution of prices in the final state is given by p, S_u, and S_d, but p does not matter. Instead a distribution determined by q, S_u, and S_d is the one that matters. If effect, for pricing purposes, we should overwrite the "up" probability with q.

Let us now take a multi-period model built for parameters μ and σ where we replace the "up" probability with q **at every step**. We know from the one-period model that the price (and return) distribution will change, but what will it be?

It turns out that the final return distribution will be still normal (and the price distribution will be log-normal), but the expected return $\widetilde{\mu}$ will be different from μ (remarkably, the variance of the stock price return is not changed):

$$\begin{aligned} E[r_T] &= \widetilde{\mu}T = \left(r - \frac{1}{2}\sigma^2\right)T \\ Var[r_T] &= \sigma^2 T \end{aligned}$$

[4] could think about adding a constant $\log U$ or $\log D$ to the **return** of the stock price computed between the current and the initial state (node) at each "up" or "down" transition, respectively.

[5] Recall that n is the number of intervals into which we divide the interval $[0, T]$.

Neither U, nor D, nor q is dependent on μ:

$$\begin{aligned} U &= \exp\left[\left(r - \frac{1}{2}\sigma^2\right)\Delta + \sigma\sqrt{\Delta}\right] \\ D &= \exp\left[\left(r - \frac{1}{2}\sigma^2\right)\Delta - \sigma\sqrt{\Delta}\right] \\ q &= \frac{\exp\left(\frac{1}{2}\sigma^2\Delta + \sigma\sqrt{\Delta}\right) - 1}{\exp(2\sigma\sqrt{\Delta}) - 1} \end{aligned} \tag{1}$$

Also, note that $\lim_{\Delta\to 0} q = \frac{1}{2}$.

Thus for the purposes of pricing we do not need to worry about the true distribution of prices. More, the price distribution is not dependent on the expected return per unit of time μ. This is good, because estimating μ is difficult in practice. The price distribution depends on the return on the money market account r, on the true volatility of the stock σ (which is easier to estimate), and on the length of the subintervals Δ. As Δ becomes smaller, the approximation of the target normal distribution becomes better and better.

It turns out that the volatility estimated from historical stock prices can not be used to accurately predict market prices of options. This is not unreasonable, since market prices might embed information that can not be inferred only from past history. In other words, the market might implicitly predict a forward looking volatility, not a backward looking one. If we accept this point of view, then we can solve an inverse problem and determine the implied volatility that makes the computed price of an option equal to its observed market price. If our models are correct, the implied volatility should be constant for all options whose underlying stock is the same, irrespective of the option's expiration date and strike price.

This, however, is not the case - the implied volatility depends, in general, both on the expiration date and the strike price of the option. This dependence is captured by the notion of implied volatility surface.[6] The volatility surface can be used to determine the implied volatility, and thus the "correct" price of options that have not been used in the computation of the surface.

0.2 Applications

After all these discussions, let us consider a few examples that will better illustrate our pricing techniques.

We denote the stock price by S, the payoffs in the final state by X, and the value of the replicating portfolio by V. We will use subscripts to denote the state for which the respective values are given.

[6] There is a small number of points where we can sample the volatility surface, as for each stock there are only a few expiration dates and strike prices. While we have not discussed this, the determination of a smooth volatility surface is an interesting problem in its own right.

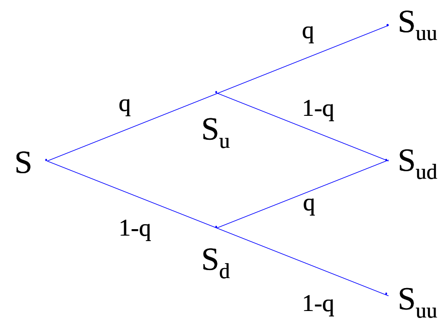

Figure 2: Two-period binomial model for European puts and calls.

0.2.1 European Calls and Puts

Consider a two-period binomial model in which we want to value a European contract having an arbitrary payoff X at time 2Δ. The lattice showing the evolution of stock prices is shown in figure 2. Note that the figure shows the equivalent martingale probabilities q.

The value of the replicating time-Δ portfolio in state u and d is equal to $V_u = e^{-r\Delta}\left[qX_{uu} + (1-q)X_{ud}\right]$ and $V_d = e^{-r\Delta}\left[qX_{ud} + (1-q)X_{dd}\right]$, respectively. The value of the time-0 replicating portfolio that has values V_u and V_d, in states u and d, respectively, is given by $V = e^{-r\Delta}\left[qV_u + (1-q)V_d\right]$. Replacing V_u and V_d in the last formula with their previously computed values, we get the following:

$$V = e^{-2r\Delta}\left[q^2 X_{uu} + 2q(1-q)X_{ud} + (1-q)^2 X_{dd}\right]$$

Employing an idea that we have encountered before, we can rewrite the value of V as the discounted time-2Δ expectation of the payoff X (when using probability q):

$$V = e^{-2r\Delta}\mathbb{E}_q[X] = \mathbb{E}_q[e^{-2r\Delta}X]$$

Until now, we have not been specific about the specific form of the payoff X. If, however, the payoff is a European call, then we have that $X_s^{call} = \max(S_s - K, 0)$, where subscript s denotes an arbitrary final state. Similarly the payoff of a European put is equal to $X_s^{call} = \max(K - S_s, 0)$. We then get:

$$\begin{aligned} V^{call} &= e^{-2r\Delta}\mathbb{E}_q[\max(S_s - K, 0)] \\ V^{put} &= e^{-2r\Delta}\mathbb{E}_q[\max(K - S_s, 0)] \end{aligned}$$

We can generalize these ideas for an n-period lattice:

$$\begin{aligned}
V &= e^{-nr\Delta}\sum_{k=0}^{n}\binom{n}{k}q^k(1-q)^{n-k}X_{\underbrace{uu...u}_{k\ times}\underbrace{dd...d}_{n-k\ times}} \\
V^{call} &= e^{-nr\Delta}\sum_{k=0}^{n}\binom{n}{k}q^k(1-q)^{n-k}\max(S_{\underbrace{uu...u}_{k\ times}\underbrace{dd...d}_{n-k\ times}}-K,0) \\
&= e^{-nr\Delta}\sum_{k=0}^{n}\binom{n}{k}q^k(1-q)^{n-k}\max(SU^kD^{n-k}-K,0) \\
V^{put} &= e^{-nr\Delta}\sum_{k=0}^{n}\binom{n}{k}q^k(1-q)^{n-k}\max(K-S_{\underbrace{uu...u}_{k\ times}\underbrace{dd...d}_{n-k\ times}},0) \\
&= e^{-nr\Delta}\sum_{k=0}^{n}\binom{n}{k}q^k(1-q)^{n-k}\max(K-SU^kD^{n-k},0)
\end{aligned}$$

Let us focus on calls. It is clear that for all values of k, such that $SU^kD^{n-k} \leq K$, the time-n payoff of the call will be 0. Let $k_* \leq n$ be the smallest value such that $SU^{k_*}D^{n-k_*} > K$. If no such value exists, then the strike price is so high that the simulated stock price never reaches it, so the call will never be exercised. If, however, k_* exists, then we can rewrite the value of the call as follows:

$$\begin{aligned}
e^{nr\Delta}V^{call} &= \sum_{k=k_*}^{n}\binom{n}{k}q^k(1-q)^{n-k}\max(SU^kD^{n-k}-K,0) \\
&= S\sum_{k=k_*}^{n}\binom{n}{k}(qU)^k[(1-q)D]^{n-k} - K\sum_{k=k_*}^{n}\binom{n}{k}q^k(1-q)^{n-k}
\end{aligned}$$

If we denote $\frac{qu}{e^{r\Delta}}$ by b, simple algebra proves that $\frac{(1-q)D}{e^{r\Delta}} = 1-b$. We then get:

$$V^{call} = \underbrace{S\sum_{k=k_*}^{n}\binom{n}{k}b^k(1-b)^{n-k}}_{A} - K\underbrace{e^{-nr\Delta}\sum_{k=k_*}^{n}\binom{n}{k}q^k(1-q)^{n-k}}_{B}$$

It might seem at first sight that the value of a call of any strike can be computed simply as $V^{call}(K) = A - KB$, where quantities A and B can be precomputed, and are independent of K. This is not true, as A and B depend on the strike price through k_*.

0.2.2 American Puts and Calls (No Dividends Case)

For simplicity, let us assume that the underlying stock does not pay dividends. To illustrate the main ideas, it is sufficient to use the two-period binomial model in figure 2.

Consider state u. In this state we have the choice of exercising the call, and get an instant payoff of $P_u = S_u - K$, or we can delay the decision of exercising the option until after the second step (i.e. until one of the possible final states has been reached). The time-Δ value of the replicating portfolio for the payoffs in states uu and ud is

$$\begin{aligned} V_u &= e^{-r\Delta}\left[qX_{uu} + (1-q)X_{ud}\right] \\ &= e^{-r\Delta}\left[q\max(S_{uu}-K,0) + (1-q)\max(S_{ud}-K,0)\right] \end{aligned}$$

It is clear that the value of the option in state u is $\max(P_u, V_u)$[7]. The condition for early exercise is equivalent to $P_u \geq V_u$, i.e.

$$S_u - K \geq e^{-r\Delta}\left[q\max(S_{uu}-K,0) + (1-q)\max(S_{ud}-K,0)\right]$$

Let us assume that both S_{uu} and S_{ud} are greater than K. We get

$$\begin{aligned} S_u - K &\geq e^{-r\Delta}\left[qS_uU + (1-q)S_uD - K\right] \\ K(1-e^{-r\Delta}) &\leq S_u[1 - e^{-r\Delta}(qU + (1-q)D)] \\ 1 &\leq e^{-r\Delta} \end{aligned}$$

The last condition is clearly impossible (if $r > 0$), so the call will not be exercised early under these assumptions.[8]

Let us now consider a put, whose payoff, if exercised at time Δ, is equal to $K - S_u$. The value of the replicating portfolio for the time-2Δ payoff is

$$V_u = e^{-r\Delta}\left[q\max(K-S_{uu},0) + (1-q)\max(K-S_{ud},0)\right]$$

Could this call be exercised early? If yes, the condition $P_u \geq V_u$ must hold:

$$K - S_u \geq e^{-r\Delta}\left[q\max(K-S_{uu},0) + (1-q)\max(K-S_{ud},0)\right]$$

Let us assume now that both S_{uu} and S_{ud} are less than K. We get

$$\begin{aligned} K - S_u &\geq e^{-r\Delta}\left[K - qS_uU - (1-q)S_uD\right] \\ K(1-e^{-r\Delta}) &\geq S_u[1 - e^{-r\Delta}(qU + (1-q)D)] \\ 1 &\geq e^{-r\Delta} \end{aligned}$$

The last condition is always true (if $r > 0$), which means that the put will always be exercised under these assumptions. Why? Here is the intuition: Because the stock price will not get over the strike in the second time period, we can see the put as being equivalent to a money market deposit that will pay K dollars at the end of the second interval, and a unit of stock sold short. The time-Δ value of the portfolio is $Ke^{-r\Delta} - S_u$. But if we exercise at time Δ we get $K - S_u$! It is better to exercise early in this case

[7] The holder will chose the decision that produces the highest value.

[8] We have not covered all the cases.

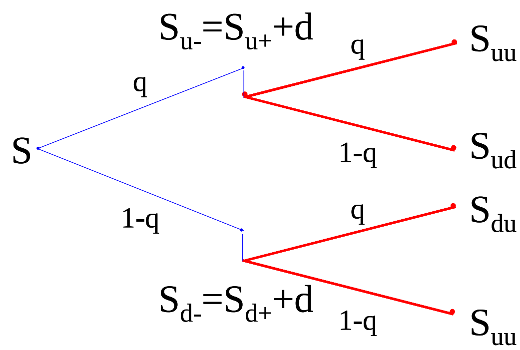

Figure 3: Two-period binomial model with discrete dividends.

because we do not have to wait to cash in on the interest that will be paid on the money market account. Are there cases in which early exercise is not as unambiguous?

Let us now focus on the time-0 value of the replicating portfolio of an American put. Using the notation above, the value of the replicating portfolio at time Δ will be $\max(P_u, V_u)$ or $\max(P_d, V_d)$, depending on whether we will reach state u or state d. The time-0 value of the replicating portfolio will be

$$V = \max(K - S, e^{-r\Delta}(q \max(P_u, V_u) + (1 - q) \max(P_d, V_d))$$

In other words, we compute the payoff if the option is exercised at time-0, and compare it to the value of the replicating portfolio at time Δ (assuming that the put has not been exercised at time 0).

This treatment can easily be generalized to more intervals and/or to other American-type payoffs.

0.2.3 American Puts and Calls (Discrete Dividends Case)

Consider the two-period binomial model represented in figure 3, and assume that we want to determine the time-0 value of an American call with strike price K and expiration date 2Δ.

To solve this problem we have to determine when, in relation to the possibility of exercising the option, will dividend payments occur. Let us assume that if the holder exercises the call at time Δ, then the holder also receives the dividend.

The value of the replicating portfolio in state u, assuming that the option is not exercised, is

$$\begin{aligned} V_u &= e^{-r\Delta}[q \max(S_{uu} - K, 0) + (1 - q) \max(S_{ud} - K, 0)] \\ &= e^{-r\Delta}[q \max(S_{u+}U - K, 0) + (1 - q) \max(S_{u+}D - K, 0)] \\ &= e^{-r\Delta}[q \max((SU - d)U - K, 0) + (1 - q) \max((SU - d)D - K, 0)] \end{aligned}$$

The payoff in state u, if the option is exercised early, will be

$$P_u = S_{u-} - K = SU - K$$

The value of the call in state u will be $\max(V_u, P_u)$. Similarly, the value of the option in state d will be $\max(V_d, P_d)$. The time-0 value of the replicating portfolio will be

$$V = e^{-r\Delta}[q \max(V_u, P_u) + (1-q) \max(V_d, P_d)]$$

0.2.4 Replicating Portfolios

In the discussion above we have not insisted on determining the replicating portfolios. We have given formulas for determining these replicating portfolios for the one-period binomial model. The number of units of stock and money market depended, in that formulation, on the payoff at time t. In the multi-period case, and for intermediate steps, the payoffs must be replaced with the value of the contract (i.e. the value of the replicating portfolio) in the current "up" and "down" states; otherwise the formulas can be employed as given.

Care must be taken, however, to use the right stock price in the case of discrete dividend payments. In figure 3, for example, one must use price S_{u+} as the stock price in state u when determining the composition of the replicating portfolio that will reproduce the payoff at time 2Δ in states uu and ud. This is because the dividend has already been paid on the stock; if we did not exercise the option, we already lost the dividend.

Consider now an intermediate node in a multi-step lattice. Assuming that you want to reproduce the payoff of an option using a replicating portfolio, you have "arrived" in the current node by holding just such a portfolio (the "incoming" portfolio). Unless your option expires or is exercised in the current node, you will also have to set up an "outgoing" replicating portfolio (otherwise you will not be able to continue tracking the evolving value of your option). The composition of the incoming portfolio will be, in general, different from that of the outgoing portfolio. An important question is whether the incoming portfolio will have enough value to cover any payments due in the current node (e.g. for discrete dividends), and to have enough left to buy the outgoing portfolio. If this property holds, and there are no transaction costs, we do not need to invest additional money into our intermediate portfolios; the initial investment will suffice until we reach a final state. Such portfolios are called "self financing."

So are our portfolios self-financing? Yes. Can you justify why?[9]

[9] While developing your answer, think of simple cases first, where the answer is (almost) obvious!

0.3 A Note on the Inaccuracy of Floating-Point Arithmetic

You might know that computations with numbers in floating-point representation[10] are only approximate. While arithmetic performed with integer types[11] is always accurate, in general, floating-point operations (e.g. addition!) are not even commutative. These issues have a major impact on the accuracy of numerical computations, and are discussed extensively in Numerical Analysis courses. We can not treat the topic here except in order to issue a few warnings.

It is possible, when using floating point representation for the sum of two positive quantities a and b to be equal to, say, a. Here is a simple Matlab program that illustrates this point:

```
>>i = 1; while(i + 1 > i) i = i + i;end; i

i = 9.007199254740992e+015
```

We will not explain here what causes the logically infinite loop to break, except to point out that for the printed value of i, the relationship $i + 1 = i$, holds in floating-point representation. Similar effects occur whenever the difference in magnitude between two numbers that are added (or subtracted) is of approximately 10^{15} or more.

Occasionally you might experience subtle effects that are due to this problem. Let us assume that you want to obtain a very precise numerical result, and your result is computed as a sum of a very large number of terms. The previous example shows that you must pay attention to the order in which you add up the numbers, especially if there are big order-of-magnitude differences between the terms, and/or the final sum will be many orders of magnitude greater than the individual terms.

So what can you do to avoid losing precision by performing many additions that have no effect because you add to a partial sum already much bigger than the term you are adding? You should compute a second partial sum by adding together the small numbers; then add up the two (or more) partial sums.

The example below illustrates this idea:

```
s = 0;
for j = 1:N; s = s + 1; end
i + s

>> i = 1e16;
>> N = 1000000;

for j = 1:N; i = i + 1; end
```

[10] The floating-point representation is the most widespread encoding of (a subset of) the real numbers in modern computers.

[11] There are special encodings suitable for integers only. Integers can also be represented in floating-point encoding.

```
i

ans = 1.000000000000000e+016


s = 0;
for j = 1:N; s = s + 1; end
i + s

ans = 1.000000000100000e+016
```

Note that the second method computed the mathematically correct sum, while the first one failed completely.