Project: Option Pricing

In a time discrete financial market models we describe the evolution of a stock price in the time interval $[0, T]$ via the stochastic difference equation

$$S_k - S_{k-1} = r \cdot S_{k-1} \cdot \Delta t + \sigma \cdot S_{k-1} \cdot Y_k \cdot \sqrt{\Delta t} \qquad (k = 1, 2, \ldots, n), \qquad S_0 = s_0.$$

Here, n is the number of periods, $\Delta t = T/n$ the time step, s_0 the initial stock price, and S_k the stock price at time $k \cdot \Delta t$. The real valued random variables $Y_1, Y_2, \ldots, Y_n$ are iid with

$$E[Y_k] = 0 \qquad \text{und} \qquad \mathrm{Var}[Y_k] = 1.$$

The "drift" r and the "volatility" σ are positive constants.

One can interprete the equation above as a time discretization of the SDE underlying the Black-Scholes model. For the random variables Y_k one can choose different distributions corresponding to different time-discretizations of the SDE

Binomial model

In the simplest case, Y_k takes values +1 und -1 with probability 1/2. Then the stock price changes in each step with probability 1/2 by a factor

$$u = 1 + r \cdot \Delta t + \sigma \cdot \sqrt{\Delta t} \qquad \text{resp.}$$
$$d = 1 + r \cdot \Delta t - \sigma \cdot \sqrt{\Delta t} \qquad .$$

Discrete Black-Scholes model

A better discretization of the SDE is obtained, if Y_k are standard normal random variables, i.e.

$$P[a \le Y_k \le b] = \int_a^b \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \, dx \qquad \text{for all } a, b \in \mathbb{R} \text{ with } a \le b.$$

Standard normal samples can be obtained in *Mathematica* via **RandomReal[NormalDistribution[0, 1]]**.

Options

An (European) option gives the owner the right to buy or sell the underlying stock at a given time for a fixed price. Its payoff is the value V at time and depends on the evolution of the stock price, i.e.

$$V = V(S_0, S_1, \ldots, S_n)$$

European call option

This option gives the right to buy the stock at time n at the price K. The payoff is $S_n - K$, if $S_n > K$, and 0 otherwise, i.e.

$$V = \max(S_n - K, 0).$$

European up-and-out-barrier option

This otption has the additional restriction that it pays zero if the stock price has exceeded level B. Its payoff is thus

$$V = \max(S_n - K, 0) \cdot I_{\{S_l \le B \text{ for all } l = 0, 1, \ldots, n\}}.$$

Fair option price

We have seen that the required that the model should and the price of the option should not create arbitrage opportunities allows to calculate the option price v_0 at time 0 as follows:

$$v_0 = (1 + r\,\Delta t)^{-n} \cdot E[V]$$

Here $E[V]$ is the expectation under the equivalent martingale measure.

Binomial model

Here the fair price can be calculated exactly. We have

$$S_k = s_0 \cdot u^{N_k} \cdot d^{k-N_k} \qquad (k = 0, 1, \ldots, n), \tag{1}$$

where $N_k = |\{1 \le l \le k : Y_l = 1\}|$ is the number of upward jumps during the first $k$steps. For i=0,1,2,..,k set

$$v_{k,i} = (1 + r\,\Delta t)^{-(n-k)} \cdot E[V \mid N_k = i].$$

Thus $v_{k,i}$ is the fair option price at time k, if the stock price at this time equals $S_k = S_0 \cdot u^i \cdot d^{k-i}$. In particular, $v_{0,0} = v_0$ is the fair price at time 0 we are looking for and

$$v_{n,i} = \max\left(s_0 \cdot u^i \cdot d^{n-i} - K, 0\right) \qquad (i = 0, 1, \ldots, n). \tag{2}$$

We have the recursion

$$v_{k-1,i} = \tfrac{1}{2}(v_{k,i} + v_{k,i+1})/(1 + r\,\Delta t) \qquad (k = n, n-1, \ldots, 1). \tag{3}$$

Equations (2) and (3) give the following algorithm to calculate v_0:

```
BinomialEuropeanCall[s0_, K_, r_, sigma_ , T_, n_] :=
  Module[{dt, u, d, q, BinomTree , value},
    dt = T / n;
    u = 1 + r * dt + sigma * Sqrt[dt];
    d = 1 + r * dt - sigma * Sqrt[dt];
    q = 1 / 2;
    BinomTree = Table[Max[s0 * u ^ i * d ^ (n - i) - K, 0], {i, 0, n}];
    Do[
      BinomTree =
        Table[(1 / (1 + r * dt)) * {q, 1 - q}.{BinomTree [[i + 2]], BinomTree [[i + 1]]},
          {i, 0, k}],
      {k, n - 1, 0, -1}];
    value = BinomTree [[1]];
    Clear[BinomTree ];
    value]
```

Similarly, one can calculate the fair price of an up-and-out-barrier option

Discrete Black-Scholes model

Here an exact computation is no longer possible but ther are several numerical methods to approximate the expaction value. One possibility are Monte Carlo methodes: One simulates N samples $\left(s_0^{(j)}, s_1^{(j)}, \ldots, s_n^{(j)}\right)$, $j = 1, \ldots, N$, of the stock price evolution, and estimates the expaction $\vartheta = E[V]$ via the arithmetic mean

$$\hat{\vartheta}_N = \tfrac{1}{N} \textstyle\sum_{j=1}^{N} v^{(j)}$$

of the corresponding payoffs $v^{(j)}$. In simple cases (e.g. european call options) Monte Carlo methods are not the most efficient, but they easily generalize to more exotic options where other methods fail.

Excercises

1. **Option pricing in the binomial model**

   1.1. Use the algorithm above to plot the evolution of the fair price of a European call option for K=100, r=0.05, σ=0.3, T=7/365 and n=40 and starting values s_0 between 90 and 120. Why is this evolution not so suprising?

   1.2. To analyze the effect of the number of time steps on the price, plot the evolution of the fair price with the above parameters for n between 20 and 140 and between 20 and 1200. How could you get a mor accurate result with less time steps?

   1.3. To calculate the price of an up-and-out-barrier option, consider for k=n,n-1,...,0 the values
   $$v_{k,i} = (1 + r\,\Delta t)^{-(n-k)} \cdot E[V_k \mid N_k = i] \qquad (i = 0,\ 1,\ \ldots,\ k),$$
   with
   $$V_k := \max\,(S_n - K,\ 0) \cdot I_{\{S_l \leq B \text{ for } l = k, k+1, \ldots, n\}}\,.$$
   Show the recursion
   $$v_{k-1,i} = \begin{cases} \frac{1}{2}\,(v_{k,i} + v_{k,i+1}) / (1 + r\,\Delta t) & \text{if } s_0 \cdot u^i \cdot d^{k-i} \leq B \\ 0 & \text{if } s_0 \cdot u^i \cdot d^{k-i} > B \end{cases}$$
   with initial condition
   $$v_{n,i} = \begin{cases} \max\left(s_0 \cdot u^i \cdot d^{n-i} - K,\ 0\right) & \text{if } K < s_0 \cdot u^i \cdot d^{n-i} < B \\ 0 & \text{else.} \end{cases}$$
   Modify the algorithm above to get a *Mathematica* funktion **BinomialEuropeanUO[s0_, K_,B_, r_, sigma_, T_, n_]**, that calculates the price of the european up-and-out-barrier option.

   1.4. Plot the evolution of the price for K=100, B=120, r=0.05, σ=0.3, T=7/365 and $n = 40$ resp. $n = 200$ with s_0 between 90 and 120. Why does the price increase up to a certain value and then starts to fall?

   1.5. In practice, it is important to calculate the “greeks”, i.e. the derivative of the option price w.r.t. the parameterss_0, r, σ... accurately in order to hedge the option. Why is the binomial model not very suitable to calculate the “Delta”, i.e. the derivative wrt. s_0 for the up-and-out call?

2. **Option pricing in the discrete Black-Scholes model**

   2.1. Simulate $N = 50$ stock price evolutions with the parameters from1.4 with $s_0 = 100$ and $s_0 = 115$. and plot them.

   2.2. Write *Mathematica* funktiones **EuropeanCallMC[s0_, K_, r_, sigma_, T_, n_, N_]** and **EuropeanUOMC[s0_, K_, B_,r_, sigma_, T_, n_, N_]**, generating a list $\left\{v^{(1)},\ \ldots.,\ v^{(N)}\right\}$ of payoffs for european call and up-and-out barrier options for N simulated stock price evolutions. (Mind the memory efficency, you do have to store the full evolution of the stock price!)

   2.3. Calculate Monte-Carlo estimators $\hat{\vartheta}_N$ for the expectation $\vartheta = E[V]$ and $\hat{\sigma}_N$ for the variance $\sigma(V)$. As Monte-Carlo estimator for the variance you can take
   $$\hat{\sigma}_N = \sqrt{\frac{1}{N-1} \sum_{j=1}^{N} \left(v^{(j)} - \hat{\vartheta}_N\right)^2}$$

Calculate the estimated fair option price v_0 for the parameters above and $N = 10\,000$. Compare to the results for the binomial model.

2.4. For the same parameters, calculate Monte-Carlo estimators for

$$\text{Delta} := (1 + r\,\Delta t)^{-n} \cdot \frac{\partial E[V]}{\partial s_0}$$

For this, simulate price evolutions $v^{(j)}$ and $w^{(j)}$ with start s_0 and $s_0 + h$ (e.g. $h = 0.1$) *using the same random numbers*, and use the estimator

$$\frac{1}{N} \sum_{j=1}^{N} \frac{w^{(j)} - v^{(j)}}{h}$$

for the difference quotient.