Homework 11: Solutions

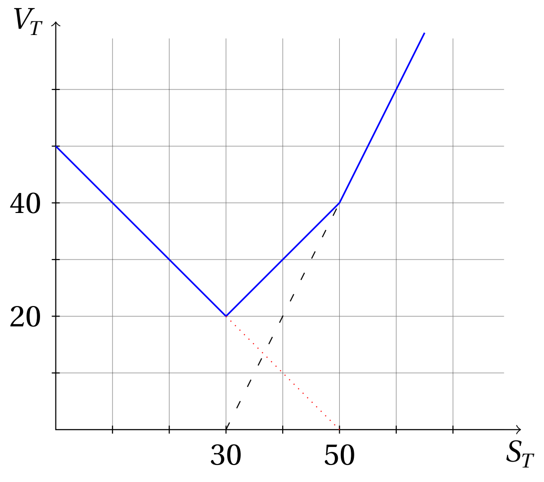
Problems to turn in individually

Problem 1

We are told that $\sigma = 0.2$, $S_0 = 30$, $r = 0.05$, and $T = 1/2$. The payoff of the option is

$$V_T = \begin{cases} 50 - S_T & \text{if } S_T < 30 \\ S_T - 10 & \text{if } 30 \le S_T \le 50 \\ 2(S_T - 30) & \text{if } 50 < S_T. \end{cases}$$

It helps to draw the graph of V_T versus S_T. Observe that there are kinks at $S_T = 30$ and $S_T = 50$ where the slope changes from -1 to $+1$, and 1 to 2, respectively. Also there is a "cash offset" of 20; in other words, if we translate the whole graph down by 20 it just touches the S_T-axis. This tells us that we could perhaps express $V_T - 20$ in terms of the payoffs of simpler objects like calls and puts (with appropriate strike prices).



Method 1: Decompose the payoff of the option as

$$V_T = 20 + \underbrace{(30 - S_T)^+}_{\text{put at 30}} + \underbrace{(S_T - 30)^+}_{\text{call at 30}} + \underbrace{(S_T - 50)^+}_{\text{call at 50}}. \tag{1}$$

By a direct application of the “no-arbitrage” principle, we conclude that the price of the left hand side in (1) should equal the price of the right hand side at all prior times $t < T$. (This is the *Law of One Price*). In particular, at $t = 0$, we have the desired answer

$$\begin{aligned} V_0 &= 20e^{-rT} + P_{\mathrm{BS}}(S_0, 30, T, r, \sigma) + C_{\mathrm{BS}}(S_0, 30, T, r, \sigma) + C_{\mathrm{BS}}(S_0, 50, T, r, \sigma) \\ &= 19.506 + 1.3259 + 2.0666 + 4.1 \times 10^{-4} = 22.9. \end{aligned}$$

Here is a simple Octave function to compute the above prices:

```
%%%%%%%%%%%%%% Given all inputs, this function returns the
%%%%%%%%%%%%%% Black-Scholes price of an option.

function price = BS (CallPutFlag,S,K,t,T,r,sigma,div);
  d1 = ( log(S/K) + (r-div+ sigma^2 / 2)*(T-t)) / (sigma*sqrt(T-t));
  d2 = d1 - sigma*sqrt(T-t);
  %%%%%%%%%%%%%%% check whether the option is a Call or a Put
  if CallPutFlag == "c"
       price = S*exp(-div*(T-t))*normcdf(d1) - K*exp(-r*(T-t))*normcdf(d2);
  else
       price = K*exp(-r*(T-t))*normcdf(-d2) - S*exp(-div*(T-t))*normcdf(-d1);
  end%if
  %%%%%%%%%%%%%%%
end%function
```

Thus, BS($"c", S_0, 30, 0, T, r, \sigma, 0$) returns 2.0666, while BS($"p", S_0, 50, 0, T, r, \sigma, 0$) returns 18.766.

Method 2: We could also have decomposed the payoff of the option as

$$V_T = \underbrace{(50 - S_T)^+}_{\text{put at 50}} + \underbrace{2(S_T - 30)^+}_{\text{2 calls at 30}} \tag{2}$$

which implies, by the Law of One Price, that

$$\begin{aligned} V_0 &= P_{\mathrm{BS}}(S_0, 50, T, r, \sigma) + 2C_{\mathrm{BS}}(S_0, 30, T, r, \sigma) \\ &= 18.766 + 2 \times 2.0666 = 22.9. \end{aligned}$$

Notice that we did not need μ, the expected annual return of the stock, in the entire problem.

Here is a simple R function to compute the above prices:

```
#Black-Scholes model R code
bs = function(s,k,t,v,r,d,cp)
{
    d1 = (log(s/k)+(r-d+0.5*v^2)*t)/(v*sqrt(t))
    d2 = d1 - v*sqrt(t)
    if (cp==1) {
        optval = s*exp(-d*t)*pnorm(d1)-k*exp(-r*t)*pnorm(d2)
    }
    else {
        optval = -s*exp(-d*t)*pnorm(-d1)+k*exp(-r*t)*pnorm(-d2)
    }
    optval
}
```

Problem 2

CRR binomial tree parameters: With quarterly intervals, the up-shift parameter is $u = e^{\sigma\sqrt{h}} = e^{0.2\sqrt{0.25}} = e^{0.2\times 0.5} = e^{0.1} = 1.1052$. The down-shift parameter is $d = 1/u = e^{-0.1} = 0.90484$. We are told that the (annual) interest rate is $r = 0.03$, thus the compounded value of \$1 in one quarter will be $1\times(1+\frac{r}{4}) = 1+\frac{0.03}{4} = R$. The risk-neutral probability of an up-shift is

$$q = \frac{R-d}{u-d} = \frac{(1+0.03/4)-e^{-0.1}}{e^{0.1}-e^{-0.1}} = 0.51246.$$

Notice that we do not need that the stock return and interest rates be on the same compounding frequency.

Problems to turn in as a group

Problem 1

Stock paying continuous dividends (contd. from HW 10): We must solve the PDE

$$C_t + (r-D)SC_S + \frac{1}{2}\sigma^2 S^2 C_{SS} - rC = 0 \tag{3a}$$

$$C(S,T) = (S-K)^+ \tag{3b}$$

where the $(\cdot)^+$ notation is defined by $x^+ := \max(x,0)$. To transform it into familiar B-S form, let

$$C(S,t) = V\big(X(S,t),t\big) \tag{4}$$

where

$$\boxed{X(S,t) := Se^{-D(T-t)}} \implies X_t = DSe^{-D(T-t)},\ X_S = e^{-D(T-t)}. \tag{5}$$

Observe that, when $t = T$, we get $X(S,T) = Se^0 = S$. Hence the final condition (3b) becomes

$$V(X,T) = (X-K)^+.$$

Next we must find C_t, C_S and C_{SS} in terms of V_t, V_X and V_{XX} to put in (3). From eqs. (4) and (5),

$$\begin{aligned} C_t &= V_X X_t + V_t = V_X \cdot DSe^{-D(T-t)} + V_t \\ C_S &= V_X X_S = V_X e^{-D(T-t)} \\ C_{SS} &= V_{XX} X_S e^{-D(T-t)} = V_{XX} e^{-2D(T-t)}. \end{aligned}$$

Plugging these into the PDE (3a), we get

$$V_X DSe^{-D(T-t)} + V_t + (r-D)Se^{-D(T-t)}V_X + \tfrac{1}{2}\sigma^2 S^2 e^{-2D(T-t)} V_{XX} - rV = 0.$$

When we substitute away S, this becomes the regular Black-Scholes PDE in V!

$$\begin{gathered} \cancel{V_X D X} + V_t + (r - \cancel{D})XV_X + \tfrac{1}{2}\sigma^2 X^2 V_{XX} - rV = 0. \\ V(X,T) = (X-K)^+. \end{gathered}$$

This means we can invoke the regular Black-Scholes formula for call options:

$$V(X,t) = X\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2)$$

$$\text{where} \qquad d_1 = \frac{\ln(X/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \qquad d_2 = d_1 - \sigma\sqrt{T-t}.$$

Finally, we can substitute for V and X in order to get the answer back in terms of the original variables C and S. Also note that

$$\ln\left(\tfrac{X}{K}\right) = \ln\left(\tfrac{S}{K}e^{-D(T-t)}\right) = \ln\left(\tfrac{S}{K}\right) - D(T-t).$$

Thus, the Black-Scholes price of a call option on a stock paying continuous dividends is

$$C(S,t) = Se^{-D(T-t)}\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2)$$

$$\text{where} \qquad d_1 = \frac{\ln(S/K) + (r - D + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \qquad d_2 = d_1 - \sigma\sqrt{T-t}.$$

Problem 2

Here's a way: The hedge fund can buy a series of puts on a domestic market index in such a way that losses are removed. This assumes being able to buy domestic market index puts where the difference between the strike price and the current stock price is bigger than the price of the put. Recall that an option is like insurance and a put is essentially insurance against losses. Of course buying these puts means having less money when the market goes up. Since the market goes up 70% of the time, you will underperform the market quite a lot, which will annoy investors.

Here's another way: The hedge fund sells a lot puts on the domestic stock market that are way out of the money. That is, they only pay out money if the market falls, say 30%. This hedge fund can tout that it does better than the market year after year with the money it gets from these puts, and if they sell enough of the puts, they will not lose money: unless the market falls more than 30% in which case there's hell to pay!

I asked a friend who owns a hedge fund about this second strategy. He said two things: (1) It would be better to instead sell far out of the money calls instead of far out of the money puts because you would have the advantage of still beating the market most of the time, but when you didn't it would be because of a particularly good year, and telling an investor you had a 25% gain when the market went up 40% is generally met with far, far few screams than telling an investor you lost 60% when the market went down 40%, and (2) he asked if I wanted to go into business with him on this, and he was only half joking. I decided to (1) decline his kind offer and stay in the Math Department and (2) never invest money in a hedge fund!

Problem 3

For the Jarrow-Rudd implementation, the basic values we need to compute are

- Time interval: $h = T/n = 1/30 = 0.033333$.
- Up shift parameter: $u = e^{(r_f - 0.5\sigma^2)h + \sigma\sqrt{h}} = 1.074687$.
- Down shift parameter: $d = 0.928644$, left to the reader to verify. Note that $d \neq 1/u$.
- Risk-free Return over one period: $R = e^{r_f h} = 1.001668$. This is the compounded value of \$1 after one period, under continuous compounding. Note that we have, as an approximation, $R = 1 + r_f h + (r_f h)^2/2 + \cdots \approx 1 + r_f h$.
- Risk-neutral probability of up shift: $q = \frac{R-d}{u-d} = 0.500016$.

In this problem we are using the notation r_f for the interest rate, instead of the usual r. Building a tree and pricing an European call and put gives the following answers: the call is priced at \$18.06, and the put is priced at \$13.18.

```
%%%%%%%%%%%%%%%%%%%%% HW9, Q5 & 6. Octave code for Jarrow-Rudd binomial tree
%(Adapted from the CRR binomial tree Octave code provided by Prof. Das.)
```

```
clear all;
begin_cpu_clk = cputime ();

%FLAGS
CallPutFlag = "p";       % "c"=call, "p"=put
option = "e";            % "a"=American, "e"=European

%INPUT DATA
S = 100; K = 100; rf = 0.05; v = 0.40; T = 1; n = 30;

%BASIC SET UP
h = T/n;
R = exp(rf*h);

u = exp((rf - 0.5*v^2)*h + v*sqrt(h))    % Up shift parameter
d = exp((rf - 0.5*v^2)*h - v*sqrt(h))    % Down shift parameter
q = (R-d)/(u-d)

%%%%PREPARE STOCK TREE
stktree = zeros(n+1,n+1);
stktree(1,1) = S;
for i=2:n+1
    stktree(1,i) = stktree(1,i-1)*u;
    for j=2:i
        stktree(j,i) = stktree(j-1,i-1)*d;
    end%for
end%for

%%% TERMINAL PAYOFF   (Value at t=T)
optval = zeros(n+1,n+1);
for j=1:n+1
    if (CallPutFlag == "c")      % call
            optval(j,n+1)=max(0,stktree(j,n+1)-K);
    else                         % put
            optval(j,n+1)=max(0,K-stktree(j,n+1));
    end%if
end%for

%%PRICE OPTION BY BACKWARD RECURSION  (Value at t<T)
if (option == "a"),       %American
%%%%%%%%%%%%%%%%%%%%
  for t=n:-1:1
    for j=1:t
            optval(j,t) = (1/R)*(q*optval(j,t+1)+(1-q)*optval(j+1,t+1));
            if (CallPutFlag == "c")          % call
                       optval(j,t) = max(optval(j,t),stktree(j,t)-K); end%if
            if (CallPutFlag == "p")          % put
                       optval(j,t) = max(optval(j,t),K-stktree(j,t)); end%if
    end%for
  end%for
%%%%%%%%%%%%%%%%%%%%
elseif (option == "e")   %European
%%%%%%%%%%%%%%%%%%%%
  for t=n:-1:1
    optval(1:t,t) = (1/R)*(q*optval(1:t,t+1)+(1-q)*optval(2:t+1,t+1));
  end%for
%%%%%%%%%%%%%%%%%%%%
else
  warning ("ERROR! Please specify American or European option");
end%if

jrval = optval(1,1)
cpu_time_elapsed = cputime () - begin_cpu_clk
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

The same Jarrow-Rudd binomial tree in R is programmed as follows for a call option:

```
#Binomial tree in R

bincall = function(s,k,t,v,r,n)
{
    #BASIC SET UP
    dt = t/n
    u = exp((r-0.5*v^2)*dt + v*sqrt(dt))
    d = exp((r-0.5*v^2)*dt - v*sqrt(dt))
    RR = exp(r*dt)
    q = (RR-d)/(u-d)

    #STOCK TREE
    stkp = array(0,dim=c(n+1,n+1))
    stkp[1,1] = s
    for (i in 2:(n+1)) {
        stkp[1,i] = stkp[1,i-1]*u;
        for (j in 2:i) {
            stkp[j,i] = stkp[j-1,i-1]*d
       }
    }

    #CALL OPTION TREE
    optval = array(0,dim=c(n+1,n+1))
    for (j in 1:(n+1)) {
        optval[j,n+1] = max(0,stkp[j,n+1]-k)
    }
    for (i in n:1) {
        for (j in 1:i) {
            optval[j,i] = (q*optval[j,i+1]+(1-q)*optval[j+1,i+1])/RR
        }
    }
    optval[1,1]
}
```

How would you modify this program for a put option? How would you modify the program for an American option?

Problem 4

The American call has the same value as the European one since there are no dividends, i.e., \$18.06. The price of the American put is \$13.73; note that there is an *early exercise premium* over and above the value of the European put.

Problem 5

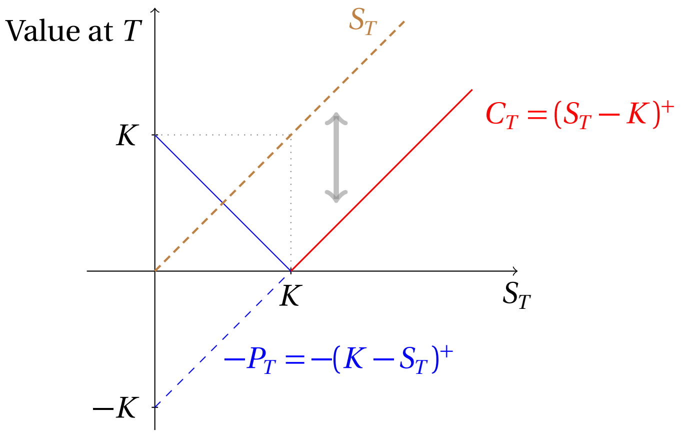
Put-call parity is

$$C - P = S - PV(K) = S - Ke^{-r_f(T-t)}.$$

It follows (by the Law of One Price) from the time-T relation

$$C_T - P_T = (S_T - K)^+ - (K - S_T)^+ = S_T - K$$

which is easy to verify graphically.



In problem 5, we got

$$C - P = 18.06 - 13.18 = 4.88$$
$$S - PV(K) = 100 - 100e^{-r_f(T-t)} = 100 - 100e^{-0.05\times 1} = 4.88.$$

Thus we have verified put-call parity.