Lab Report #6: Options & Volatility

Revised: April 8, 2013

Due at the start of class. You may speak to others, but whatever you hand in should be your own work. Please include your Matlab code.

Solution: Answers follow. See the Matlab code at the end for calculations and figures.

1. *BSM formula.* We'll examine the BSM formula in some purely theoretical numerical examples. In what follows, the current price of the underlying is 100, the option maturity is one year, and the one-year bond price is 0.98.
   (a) If volatility $\sigma = 0.10$, what are the prices of call options at strike prices of 95, 100, and 105?
   (b) What are the prices of put options with the same strikes?
   (c) If volatility rises to $\sigma = 0.125$, what happens to the prices of calls?
   (d) For strikes of 90, 100, and 110, graph the call price against volatility σ using a grid between (roughly) 0.01 and 0.50. (This gives you three lines, one for each strike.) How do call prices vary with volatility? Does the pattern vary with the strike price?
   (e) For a strike of 110 and a call price of 2.00, what is the implied value of σ?

Solution:

(a) Call prices are 8.24, 5.03, and 2.76 at strikes of 95, 100, and 105, resp.

(b) For put options, the easiest route is to use put-call parity. The put prices are 1.34, 3.03, and 5.66.

(c) Calls rise to 9.03, 6.00, and 3.74. Why rise? Options prices are increasing in σ. You can show this by differentiating the function or plotting it, as we do next.

(d) See Figure 1 of the Matlab program. You see that call prices increase with volatility in all cases. (Puts, too, for that matter.) The relation is close to linear except for very small values of σ. (Think about the value of an option as σ approaches zero.)

(e) From the values computed for the figure, $\sigma = 0.12$ is about it.

2. *Volatilities on S&P 500 E-mini options.* For S&P 500 E-mini options, the prices of options are more conveniently expressed in terms of their implied volatilities. We'll compute them here for quotes reported on March 15, 2012:

	Call Price		Put Price	
Strike	Bid	Ask	Bid	Ask
1340	82.50	85.75	28.25	30.25
1350	75.25	78.25	30.75	32.75
1360	68.00	71.00	33.25	35.50
1370	61.25	64.00	36.25	38.75
1380	54.50	57.25	39.50	42.25
1390	48.25	50.75	43.25	45.75
1400	42.25	44.75	47.00	50.00
1410	36.75	39.25	51.25	54.50
1420	31.75	33.75	56.00	59.25
1430	27.00	29.00	61.25	64.50

The price of the underlying contract was 1395.75. The interest rate was essentially zero, so the appropriate bond price was one. The options expire June 15, so $\tau = 3/12 = 1/4$.

(a) Compute "mid" quotes as averages of bid and ask. Use put-call parity to compute call prices from mid puts. Plot call prices — bid, ask, and implied by puts — against the strike. How do they compare?

(b) Write a program to compute implied volatilities for mid quotes of call options. Graph them against the strike. What shape does the resulting "smile" have? What does the shape suggest to you about the risk-neutral probabilities?

(c) Look up the current prices of June 2013 options. (The CME has current quotes on their website.) How do the prices of at-the-money options compare? What does that tell you about the market now and then?

Solution:

(a) Run the Matlab program for the figure. Call prices computed from mid puts (asterisks) are within the bid-ask spread for call prices, so evidently people in this market understand the arbitrage possibilities from violations of put-call parity.

(b) This is a more involved calculation. The idea is to use some kind of root-finding method to compute implied volatilities from call prices. In the figure plotted by the Matlab program, we see that implied vols decline with strike. That means prices at low strikes are relatively more expensive than the BSM formula with constant σ would suggest. If you did this for a broader range of maturities, you would also see some convexity in the smile that isn't apparent here.

(c) I found option prices at http://www.cmegroup.com/trading/equity-index/us-index/e-mini-sandp500.html. The underlying futures contract was trading at 1551. Call options at strikes of 1550 and 1555 traded as 35.75 and 32.75. Both are below what we saw for at-the-money options in the prices reported above, suggesting that implied volatility is lower. That's roughly what the VIX measures.

Matlab program:

```
%  hw6_s13.m
%  Matlab program for Lab Report #5, Spring 2012
%  NYU course ECON-UB 233, Macro foundations for asset pricing, Mar 2012.
%  Written by:  Dave Backus, March 2012
format compact
format short
clear all

%%
disp('Answers to Lab Report 6s')

disp(' ')
disp('-------------------------------------------------------------------')
disp('Question 1 (option prices)')

disp(' ')
disp('Inputs')
tau = 1;                 % maturity
q_tau = 0.98;           % bond price
s = 100.00;              % current price of underlying
k = [95; 100; 105];     % strike prices

% BSM formula
% define price as function of sigma, two steps for clarity
% NB:  these are what Matlab calls anonymous functions, see
% http://www.mathworks.com/help/matlab/matlab_prog/anonymous-functions.html
d = @(sigma,k) (log(s./(q_tau.*k))+tau*sigma.^2/2)./(sqrt(tau)*sigma);
call = @(sigma,k) s*normcdf(d(sigma,k)) - q_tau.*k.*normcdf(d(sigma,k)-sqrt(tau)*sigma);

disp('(a)')
q_call_10 = call(0.10,k)

disp('(b)')
% compute put prices using put-call parity
q_put_10 = q_call_10 + q_tau*k - s
```

```
disp('(c)')
% compute call prices with higher sigma
q_call_125 = call(0.125,k)

%disp(' ')
%disp('Strike, Call Prices at vol 0.10 and 0.15, Put Prices at 0.10')
%[k q_call_10 q_call_15 q_put_10]

disp('(d)')
% call price v sigma
sigma = [0.001:0.01:0.50]';

q_call_90 = call(sigma,90);
q_call_100 = call(sigma,100);
q_call_110 = call(sigma,110);

figure(1)
clf
plot(sigma,q_call_90,'k')
hold on
plot(sigma,q_call_100,'b')
plot(sigma,q_call_110,'m')
axis([0 0.5 0 25])
xlabel('Volatility')
ylabel('Call Price')
text(0.02,22,'Strikes of 90, 100, and 110 as you move down')

%print ...

disp('(e)')
disp(' ')
[dummy,i] = min(abs(q_call_110-2.00));
sigma_implied_parte = sigma(i)


%%
disp(' ')
disp('-------------------------------------------------------------------')
disp('Question 2 (implied volatilities)')
clear all
format compact

disp(' ')
disp('Inputs')
tau = 3/12
```

```
q_tau = 1.00
q = 1395.75

data = [
    1340  82.50 85.75 28.25 30.25;
    1350  75.25 78.25 30.75 32.75;
    1360  68.00 71.00 33.25 35.50;
    1370  61.25 64.00 36.25 38.75;
    1380  54.50 57.25 39.50 42.25;
    1390  48.25 50.75 43.25 45.75;
    1400  42.25 44.75 47.00 50.00;
    1410  36.75 39.25 51.25 54.50;
    1420  31.75 33.75 56.00 59.25;
    1430  27.00 29.00 61.25 64.50];

b = data(:,1);
call_bid = data(:,2);
call_ask = data(:,3);
put_bid = data(:,4);
put_ask = data(:,5);

call_mid = (call_bid+call_ask)/2;
put_mid = (put_bid+put_ask)/2;

disp(' ')
disp('(a) Calls from puts')

call_fromputs = q - q_tau*b + put_mid;

figure(1)
clf
plot(b, call_bid, 'b')
hold on
plot(b, call_ask, 'm')
plot(b, call_fromputs, 'k*')
title('Bid and Ask Call Prices')
xlabel('Strike Price')
ylabel('Call Price')
text(1350,30,'blue is bid, magenta is ask, * is from puts')

print -depsc hw5_q2a.eps

disp(' ')
disp('(b) Implied vols for mid calls via secant method')
clear functions
```

```
% BSM formula
% define f = call price as function of sigma, two steps for clarity (or not?)
d = @(sigma,b) (log(q./(q_tau.*b))+tau*sigma.^2/2)./(sqrt(tau)*sigma);
f = @(sigma,b) q*normcdf(d(sigma,b)) - q_tau.*b.*normcdf(d(sigma,b)-sqrt(tau)*sigma) ...
        - call_mid;

% convergence parameters
tol = 1.e-8;
maxit = 50;

% starting values
% NB:  we do this for log(sigma), which makes sure sigma is positive
x_before = log(0.08) + zeros(size(b));
x_now = log(0.12) + zeros(size(b));
f_before = f(exp(x_before),b);
f_now = f(exp(x_now),b);

% compute implied vol
t0 = cputime;
for it = 1:maxit
    fprime = (f_now - f_before)./(x_now - x_before);
    x_new = x_now - f_now./fprime;
    f_new = f(exp(x_new),b);
    diff_x = max(abs(x_new - x_now));
    diff_f = max(abs(f_new));
%    [it diff_x diff_f]

    if max(diff_x,diff_f) < tol, break, end

    x_before = x_now;
    x_now = x_new;
    f_before = f_now;
    f_now = f_new;
end

% display results
it
time = cputime - t0
diffs = [diff_x diff_f]
disp(' ')
disp('Strike/1000 and Vol')
vol = exp(x_new);
[b/1000 vol]

figure(2)
clf
```

```
plot(b, vol, 'b')
hold on
plot(b, vol, 'b+')
xlabel('Strike Price')
ylabel('Implied Volatility')

print -depsc hw5_q2b.eps

return
```