QF4102 Financial Modeling

Assignment 1 Report

Group 19

A0098071 Khuong Bich Ngoc
A0098100 Le Hoang Van

A1.1 European down-and-in call option

(i)–(ii) Closed form Black-Scholes (B-S) formula for European down-and-in call option

Listing 1: Matlab command

```
H = 1.3 ; X = 1.1; S0 = 1.3:0.025:(1.3+0.025*20);
r=0.03 ; q=0.01; sigma=0.25; T=0.5;
V_downin=BS_EurDownInCall( S0,X,r,T,H,sigma,q);
plot(S0,V_downin,'r-');
xlabel('Values of asset') % x-axis label
ylabel('Value of option') % y-axis label
legend('y = Value of European down and in call')
```

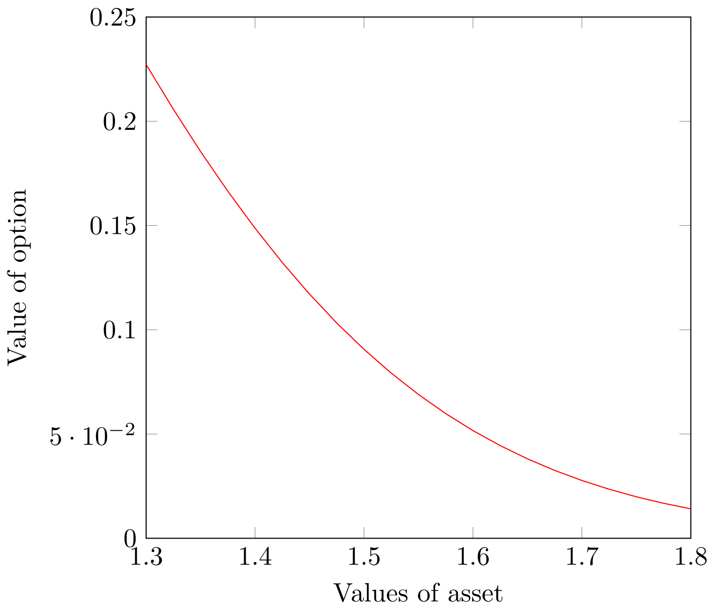

Figure 1: Values of down and in call against S_0 values

(iii) Barrier call vs. vanilla call

Listing 2: Matlab command

```
V_vanilla=BS_EurVanillaCall(S0,X,r,T,sigma,q);
plot(S0,V_downin,'r-',S0,V_vanilla,'b-');
xlabel('Values of asset') % x-axis label
ylabel('Value of option') % y-axis label
legend('Value of European down and in call','Value of European plain vanilla call')
```

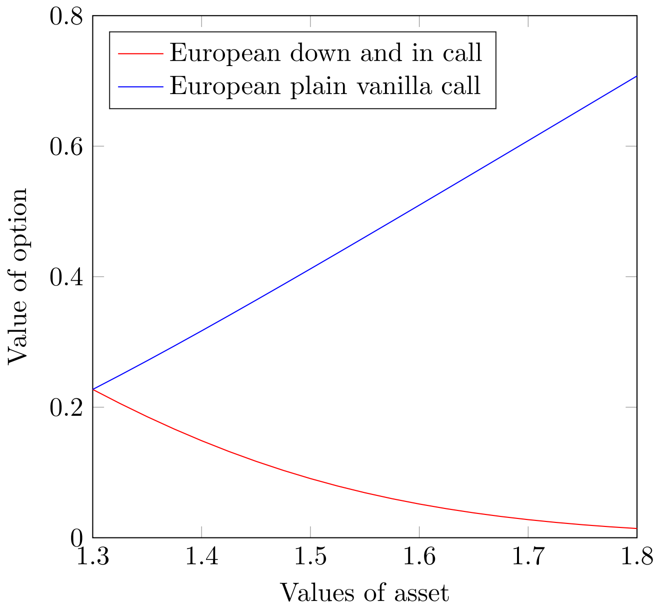

Figure 2: Graph of option values against S_0 values

The down and in call is cheaper than the vanilla call, which is intuitive because barrier options generally carry the risk of being knocked out, or not being knocked in (and hence expire worthless.)
The only time when two options have the same value is when $S_0 = H = 1.3$. As S_0 increases, or the distance between H and S_0 grows bigger, the down and in call becomes increasingly cheaper than the vanilla call. It is because as S_0 moves further away from H, the lower the probability that S_0 can hit the barrier and become worthy.

A1.2 Binomial tree method (BTM) for down-and-out European call

(i)–(ii) Binomial tree method vs. Black-Scholes

Listing 3: Matlab command

```
H=8; X=6.5; S0=8.5; sigma=0.35; q=0.02; r=0.05; T=0.5;
V_bs=BS_EurDownOutCall( S0,X,r,T,H,sigma,q)

%Ans: 0.7985

for i=1:1:100
V(i)=BTM_EurDownOutCall( S0,X,r,T,H,sigma,q,i+199);
end
plot(200:1:299,V,'b-');
hold on
plot([200 299],[V_bs V_bs],'r-');
legend('BTM','B-S','Location','northwest')
```

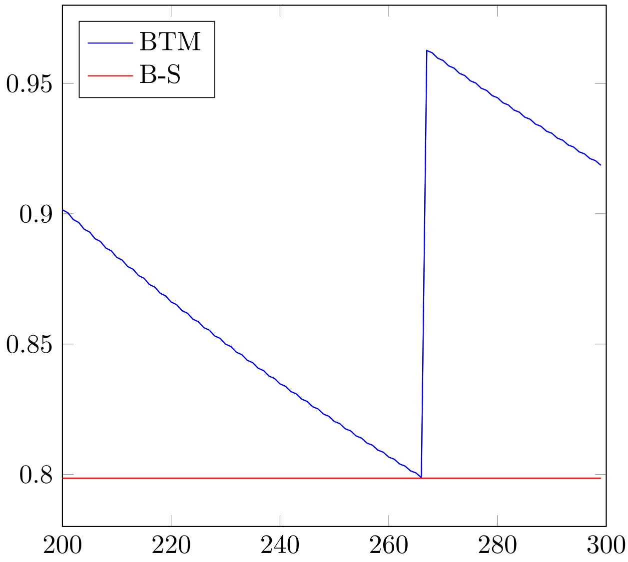

Figure 3: Graph of option values against S_0 values

We observe a zigzag convergence for the down and out call using BTM. When a larger range of N is being examined, we can see that the convergence of the usual BTM approach is actually very slow and requires a quite large number of time steps to obtain a reasonably accurate result. It is because the barrier being assumed by the tree is different from the true barrier.

Table 1: Numerical results of BTM American down and out call

$(H = 8, X = 6.5, S_0 = 8.5, \sigma = 0.35, q = 0.02, r = 0.05, T = 0.5)$

N	Price
200	2.0187
500	2.0196
1000	2.0195

(iii) As expected, American options are more expensive than their European counterparts, thanks to the compounded time value of early exercising.

A1.3 Lookback floating strike European call

(i)–(ii) At initialization, add W_0^N to the top of vector W, making it become size $N + 2$, so that boundary option values can be calculated in the same way as interior option values. At any iteration of n in the for loop, $W(1)$ will always have the same value as W_0^{n+1}, and $W(j)$ will be the calculated value of W_j^n. After we finish calculating W_j^n, $W(1) = W(2) = W_0^{n+1}$ for the next iteration of n.

Table 2: Numerical results of BTM newly issued lookback floating strike European call

$(T = 0.5, S_0 = 5, \sigma = 0.35, q = 0.01, r = 0.02,$ correct price $= 0.9189)$

N	Price
50	0.8515
500	0.8968
5000	0.9119
10000	0.9139

(iii) The BTM converges pretty fast for small values of N. For $N = 50$ we are only 7% off from the true value. However, the marginal gain in accuracy is small even if we use a very large N; it takes around 5,000 time steps to be within 2 decimal places to the true value.

(iv) The previously issued lookback floating strike option is worth more because with a running minimum lower than the current underlying price, it will be likely to expire with a lower observed minimum, thus higher payoff.

Table 3: Numerical results of BTM previously issued lookback floating strike European call

$(T = 0.5, S_0 = 5, \sigma = 0.35, q = 0.01, r = 0.02, m = 4.75)$

N	Price
50	0.8716
500	0.9151
5000	0.9299
10000	0.9319

Table 4: Numerical results of BTM previously issued lookback floating strike American call

$(T = 0.5, S_0 = 5, \sigma = 0.35, q = 0.01, r = 0.02, m = 4.75)$

N	Price		Difference
	American	European	
50	0.871624	0.871623	1.2788e-06
500	0.915062	0.915061	1.8615e-06
5000	0.929870	0.929872	2.0082e-06
10000	0.931915	0.931913	2.0263e-06

(v) The American calls are a little bit more worthy, although the difference is very small. However, this difference seems to grow as N increases.

A1.4 Newly issued European floating strike Asian arithmetic-average put option

Table 5: Computation of BTM newly issued floating strike Asian arithmetic-average put option

$(T = 0.25, S_0 = 7.5, \sigma = 0.45, q = 0.02, r = 0.03)$

N	Price	Time (s)
5	0.3776	0.004011
10	0.3781	0.009909
15	0.3790	0.043051

(i) The computation time seems to increase exponentially with N, which is not surprising because the binomial tree also grows exponentially with N. For the code, $\tau_n(k)$ returns the price state index j_n, associated with the path state index k at time t_n. To this point, we keep track of the $A's$ by converting the order of A into a binary string with 1 and 0 representing an up step and down step, respectively. Then to get j_n we simply count the occurrence of 1's in said string. This can be done by calling `(sum((dec2bin((1:2N̂)-1)-48),2)+1)`.

Table 6: Computation of FSGM newly issued floating strike Asian arithmetic-average put option

$(T = 0.25, S_0 = 7.5, \sigma = 0.45, q = 0.02, r = 0.03, \rho = 1/2)$

N	Nearest point interpolation		Linear interpolation		Quadratic interpolation	
	Price	Time (s)	Price	Time (s)	Price	Time (s)
5	0.3634	0.024773	0.3793	0.007377	0.3705	0.014260
10	0.4077	0.017661	0.3828	0.010992	0.3749	0.018823
15	0.4376	0.022043	0.3839	0.019728	0.3769	0.036613
100	0.6119	0.538865	0.3840	0.419719	0.3806	1.026534
200	0.6366	2.882865	0.3834	2.455010	0.3810	5.392594

(ii) Given the result from BTM, the true value is expected to vary around 0.37–0.38. In view of this fact, the nearest point interpolation performs worst, as the values computed clearly diverge from the true one. Both linear and quadratic seems to converge to the true value, although the latter is more consistent, which comes as a trade-off to speed because more computations are required.

A1.5 FSGM for previously issued path dependent options

(i) Previously issued American fixed strike Asian geometric average put

Table 7: Numerical results of FSGM American fixed strike Asian geometric average put

$(S_0 = 5.25, r = 0.03, q = 0.02, T = 0.25, \sigma = 0.40, a = 5.05, X = 5.90, \rho = 1/2)$

N	Nearest point interpolation	Linear interpolation	Quadratic interpolation
50	0.8696	0.9095	0.9058
100	0.8528	0.9132	0.9109
200	0.8500	0.9150	0.9132

The nearest point interpolation appears to be the worst performer – the computed values start to decrease with time while they do increase for both linear and quadratic interpolation.

(ii) Previously issued fixed trike lookback put

Table 8: Numerical results of FSGM fixed trike lookback put

$(S_0 = 1.65, r = 0.03, q = 0.02, T = 0.25, \sigma = 0.45, m = 1.55, X = 1.8)$

N	Price
50	0.4154
100	0.4199
200	0.8500

The estimated value of this lookback put is quite large compared to the asset price at present. This is partly because of compounded time value of the early exercising ability of this option, and because of the lookback feature. Time elapsed for the time steps is relatively fast, with a slight noticeable increase in the computation time when N gets higher.