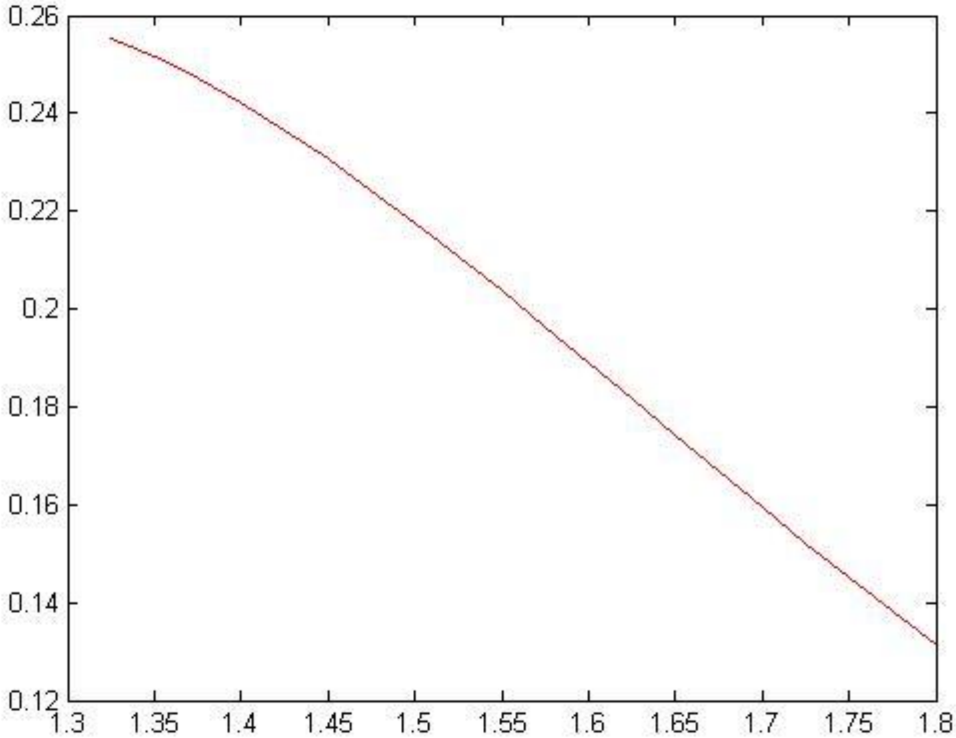
Assignment 1
G46
Joel Liew (A0004624U)
Samuel Tow Wee Yap (A0102724U)

A1.1)

i) The true value of the down and in option is at 0.7985. With the files attached we can get the answer by calling the function, downin(S,X,r,T,sigma,q,H)

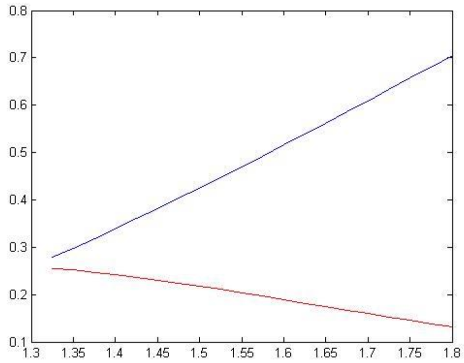
ii)
```
S=1.325:0.025:1.8
Y=downin(S,1.1,0.03,0.5,0.4,0.02,1.3)
plot(S,Y,'-r')
```



<u>Comment</u>

It shows that the value of the down and in option is decreasing in value as S is increasing. This is because S is further away from the barrier and therefore less likely for the price of the stock to hit below the barrier to initiate the call option.

iii) [Continuing from the previous code]
```
Yx=BS_call(S,1.1,0.03,0.5,0.4,0.02)
hold on
plot(S,Yx,'-b')
```



<u>Comment</u>
We can see that the vanilla call options value and the down and in call option is diverging as S0.

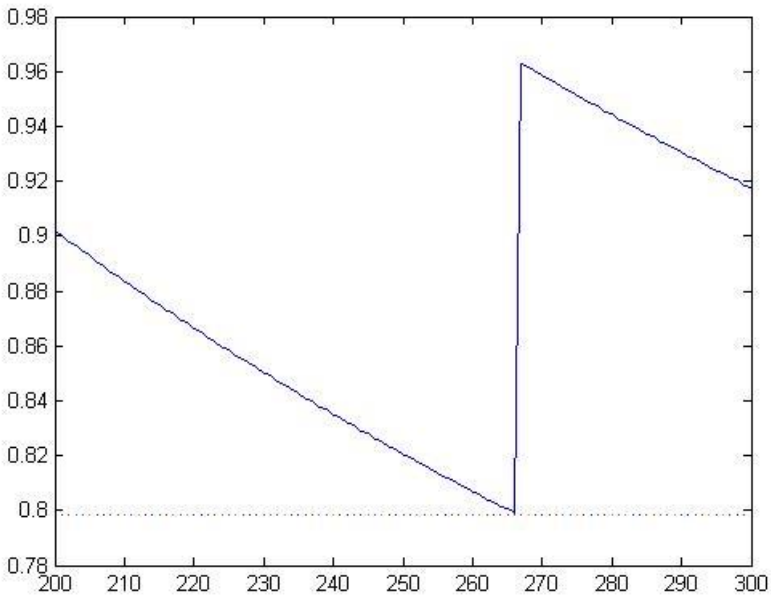
A1.2)

i) The true value of the down and out option is at 0.7985. With the files attached we can get the answer by calling the function, downout1(8.5,6.5,0.05,0.5,0.35,0.02,8).

ii) In the files attached we can get the answer for a down and out option value using the BTM method by calling the function btm_EurCall(8.5,6.5,0.05,0.5,0.35,0.02,N,8) with the N you desire.

We get the plot below by using the code:

```
for N=200:1:299
V(N-199)=btm_EurCall(8.5,6.5,0.05,0.5,0.35,0.02,N,8)
end
plot(200:299,V)
hold on
plot(200:299, downout1(8.5,6.5,0.05,0.5,0.35,0.02,8))
```



Comment

The bottom line is the true value of the option. When N=200 the point starts 0.9015 and decreases to the true value as N increases, until it reaches N=266 (whose value is 0.7987). However, it then jumps sharply to 0.9626 when N=267. After that point, the value of the options begins to approach the true value gradually again.

iii)

N	200	500	1000
Binomial Tree – American down and out call option btm_amCall(8.5,6.5,0.05,0.5,0.35,0.02,N,8)	2.0187	2.0196	2.0195

As the number of iteration increases, the value of the binomial method start to increase It then stays roughly the same after the iteration goes above 500.

A1.3)

iii)

N	50	500	5000	10000
Binomial Tree – Lookback Floating Strike European Call,t0 cfl_bm(0.02,0.01,5,0.35,0.5,N)	0.8515	0.8968	0.9119	0.9139
Closed Form – Lookback Floating Strike European Call,t0 cfl(0.02,0.01,5,5,0.35,0.5)	0.9189	0.9189	0.9189	0.9189

Comment

As the number of iterations increases, the value obtained from the binomial method converges towards the value produced by the closed formula in i). However, the convergence rate decreases as the number of iterations increases.

iv)

N	50	500	5000	10000
Binomial Tree – Lookback Floating Strike European Call,tn cfl_bm3(0.02,0.01,5,4.75,0.35,0.5,N)	0.8838	0.9189	0.9311	0.9328
Closed Form – Lookback Floating Strike European Call,tn cfl(0.02,0.01,5,4.75,0.35,0.5)	0.9369	0.9369	0.9369	0.9369

Comment

The option price obtained from the binomial method is quite close to the price obtained from the closed formula. The price could possibly be closer if a quadratic interpolation is used instead. Also, note the slowing rate of increment as the number of iterations increases.

v)

N	50	500	5000	10000
Binomial Tree – Lookback Floating Strike American Call,tn cfl_bm_am(0.02,0.01,5,4.75,0.35,0.5,N)	0.883789	0.918903	0.931083	0.932772
Binomial Tree – Lookback Floating Strike European Call,tn cfl_bm3(0.02,0.01,5,4.75,0.35,0.5,N)	0.883787	0.918901	0.931081	0.932769
Difference	1.5909e-06	1.9775e-06	2.0464e-06	2.0534e-06

Comment

The values of the American option are very close to the European ones. This could be because that there is not much additional value to exercise the option early as the payoff is: max(S-Smin,0). The Smin is the lowest corresponding stock price in the lifespan of the Option. Hence, if St=Smin, there is no value in exercising early. If St>Smin and t<T, there is only a marginal value in exercising earlier, since the Smin is already anchored in the historical low.

A 1.4)

i)

N	5	10	15
Binomial Tree – Floating Strike Asian Arithmetic-average European Put aafsput(0.03,0.02,7.5,0.45,0.25,N)	0.3776	0.3781	0.3790
Running Time (s)	0.001077	0.000706	0.004672

ii)

N	5	10	15	100	200
Linear Interpolation (Running Time, s) fsg_aafsput_linear(0.03,0.02,7.5,0.45,0.25,N,0.5)	0.3793 (0.002497)	0.3828 (0.001574)	0.3839 (0.003130)	0.3840 (0.255384)	0.3834 (1.710569)
Quadratic Interpolation (Running Time, s)	0.3705 (0.000677)	0.3749 (0.001961)	0.3769 (0.004169)	0.3806	0.3810 (2.223355)

fsg_aafsput_quad(0.03,0.02,7.5,0.45,0.25,N,0.5)				(0.347508)	
Nearest Point (Running Time,s) fsg_aafsput_near(0.03,0.02,7.5,0.45,0.25,N,0.5)	0.3634 (0.000912)	0.4077 (0.001330)	0.4428 (0.002863)	0.6130 (0.286186)	0.6368 (2.063820)

Comment

The Forward Shooting grid method gives a much more accurate result as compared to the standard binomial tree. Further, although the computation time for 15 iterations and below is longer as compared to the binomial tree, the computation time for 100 iterations and above is much shorter than that of the binomial tree.

Comparing the 3 different interpolation methods, the nearest point interpolation method gives the most inaccurate results. Between the Linear and Quadratic Interpolation, the Quadratic method seems to be more accurate, from the fact that the values are increasing across the different number of N's , whereas there is a turn from N=100 to N=200 for the Linear interpolation. Of course, the quadratic interpolation running time suffers due to the increased complexity.

A 1.5)

i)

N / Running Average Time	50 / 10	100 / 20	200 / 40
Linear Interpolation (Running Time, s) fsg_agput_am_linear(0.03,0.02,5.25,5.9,5.05,Ave_time,0.4,0.25,N,0.5)	0.9095 (0.068449)	0.9132 (0.401913)	0.9150 (2.738208)
Quadratic Interpolation (Running Time, s) fsg_agput_am_quad(0.03,0.02,5.25,5.9,5.05,Ave_time,0.4,0.25,N,0.5)	0.9047 (0.081689)	0.9108 (0.496428)	0.9131 (3.321839)
Nearest Point (Running Time,s) fsg_agput_am_near(0.03,0.02,5.25,5.9,5.05,Ave_time,0.4,0.25,N,0.5)	0.8679 (0.070135)	0.8558 (0.455559)	0.8500 (3.294832)

Comment

As usual, nearest point method gives the least accuracy among the 3 different methods. Interestingly, the time taken for the nearest point method is slower compared to the linear interpolation. Linear Interpolation gives the highest option value in all 3 cases compared to the other 2.

ii)

N / Running Average Time	50	100	200
American Fixed Strike Lookback Put (Running Time, s)	0.4157 (0.032392)	0.4215 (0.108026)	0.4232 (0.559721)

fsg_lbput_am(0.03,0.02,1.65,1.8,1.55,0.45,0.25,N)			

<u>Comment</u>

Unlike the Forward Shooting Grid with a certain interpolation method done in the previous questions, this American Fixed strike lookback put does not require any interpolation. Hence, this is why the computation times is much faster for 200 steps – comparing between A1.5 i) and this question. As for accuracy wise, the results are relatively accurate and the results slowly converges (from 100 to 200) as the number of iterations increases.