Name: Fei Liu
UNI: fl2312
Tele: 9176557918

Report for HW7

PLEASE NOTE THAT:

I am not explaining for every output because the main scheme is the same for different barriers and the difference is in how the barriers are set up and which payoff function I used. So, I am explaining the PSOR American style options in general and specify the difference in each output item separately in the end.

Part 1 Implementation of algorithm:

For European Vanilla call option, the closed formula is exactly the same as in HW1. The Black-Scholes formula is as below:

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

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} = d_1 - \sigma\sqrt{T-t}.$$

For PSOR American-style options in general, we have the Crank Nicolson Matrices before and the LUD method is implemented last time. In this homework, I did two major changes:

One is to add the PSOR method by using the below formula for each iteration.

$$v_i^{n+1} = \max\left(v_i^n + \frac{\omega}{M_{ii}}\left(q_i - \sum_{j=1}^{i-1} M_{ij} v_j^{n+1} - \sum_{j=i}^{N} M_{ij} v_j^n\right), h_i^{k+1}\right)$$

The other one is to impose barrier conditions wherever before the iteration and during the iterations. The rest of the code is essentially the same. I think this above formula can be revised by adding a (1-w) before the first term. In this way, the formula makes more sense. However, the above one is the one appeared in the slides. I am asking about it by e-mailing professor and adding this did yield better results. So, I keeps this one in my code.

For PSOR American-style options with Barrier options, I simply enforce the barrier conditions each step. The difference is DO bounds below while UO bounds above. The lines of code is exactly the same except for that the index range is different.

The LUD method is exactly the same code I used in lab6 and the difference is adding the
Formula

$$V(t+1) = \max(V(t), h(t+1))$$

Where h(t+1) is the payoff at time t+1.
The formula exercise the option when needed and carry that value on.

Part 2 Benchmark

(Note that I will put the analysis of change given different Nt, Ns, and Smax to next section)

	My Code for typical case	Haug's VBA Code/Professor's VBA code
Closed-form European Style Vanilla call option price	18.2378	18.23782
Crank Nicolson PSOR American Style vanilla call	17.9524	17.03693
Crank Nicolson PSOR American Style vanilla Put	7.5276	**N/A**
Crank Nicolson LUD American Style vanilla Put	5.9901	**5.9985 (Used American_Put_binomial as a close substitute form professor's code)**
Closed-form European Style DO call option price	16.4110	**16.41517**
Crank Nicolson PSOR American Style DO call	16.2835	**16.40286**
Crank Nicolson PSOR American Style DO Put	3.2817	**N/A (It seems this one is not available even in professor's code but LUD and PSOR are reconfirming themselves)**

Name: Fei Liu
UNI: fl2312
Tele: 9176557918

Crank Nicolson LUD American Style DO Put	3.2727	**N/A (It seems this one is not available even in professor's code but LUD and PSOR are reconfirming themselves)**
Closed-form European Style UO call option price	1.4649	**1.448421**
Crank Nicolson PSOR American Style UO call	15.4580	**N/A(The professor's VBA code only contains UO for European style option)**
Crank Nicolson PSOR American Style UO Put	6.1082	**N/A (similarly, this is not available as well.)**
Crank Nicolson LUD American Style UO call	5.3894	**N/A (similarly, this is not available as well.)**

Name: Fei Liu
UNI: fl2312
Tele: 9176557918

Part 3 Impact of change of parameters:

	My Code with Smax=400	My code with Smin= 50	My Code for typical case
Closed-form European Style Vanilla call option price	18.2378	18.2378	18.2378
Crank Nicolson PSOR American Style vanilla call	20.8086	17.9524	17.9524
Crank Nicolson PSOR American Style vanilla Put	6.5121	7.5276	7.5276
Crank Nicolson LUD American Style vanilla Put	4.9475	5.9901	5.9901
Closed-form European Style DO call option price	16.4110	16.4110	16.4110
Crank Nicolson PSOR American Style DO call	19.5425	16.2835	16.2835
Crank Nicolson PSOR American Style DO Put	1.1522	3.2817	3.2817
Crank Nicolson LUD American Style DO Put	1.1500	3.2727	3.2727

Closed-form European Style UO call option price	1.4649	1.4649	1.4649
Crank Nicolson PSOR American Style UO call	17.3082	15.4580	15.4580
Crank Nicolson PSOR American Style UO Put	4.8775	6.1082	6.1082
Crank Nicolson LUD American Style UO call	4.1114	5.3894	5.3894

It seems that the SOR mehod gives a little higher result for American style put and American style barrier put, but close in the American style call. This may different may be because the LUD method gives only first order convergence in time and the SOR method keeps the second order convergence in both time and space.

I think this θ is referring to the Douglas method. If I take it to be 0 it is explicit method and 1 to be implicit. The current value of 1/2 is stable and order 2 accurate in time and space. However, adjusting those will force me to recode the whole thing again. I am running out of time, so I give up points to do this adjustment. Rather, I just put my reasoning here and what I would expect. If θ takes 0,1,2, there should be the same truncation errors and when $\theta = \frac{1}{2} - \frac{1}{12}\Delta S^2/\Delta t$, the error should be very small. Below order should holds.

$$O(\tfrac{1}{2} - \tfrac{1}{12}\Delta S^2 - \theta\Delta t, \Delta S^4, \Delta t^2).$$

This is only a small part of the report and coding it and testing it takes too long before finals. Sorry.

I understand that SOR method is more accurate than LUD method in order of accuracy in time since LUD is only first order accurate in time. The second reason is that the matrix in LUD is changing every time as the value at each iteration may be changed to a value out of payoff function. The matrix has to change accordingly making the process slower than the European counterpart. SOR is iterating every step so the accuracy can be controlled but LUD can't be stopped and decide which accuracy to use.

The comparison with benchmark code is in earlier session. Thank you for reading my report the whole semester. Have a wonderful winter break!