Jingxia Zhu
IE 523
12/13/2018

Part 1:
This program is heavily based on Chap 9 and Chap 7 sample code.
As instructions, we need to attempt the problem using explicit and adjustment simulation. Explicit method is easy: we need to check whether each simulated path has touched the barrier or not. If the barrier is touched, the payoff should be 0. After we get the aggregate payoff for every trial, we discount back and divide by number of trials. I utilized my barrier_option() and prob_correction() function for reusability.

For adjustment simulation: As told in lecture, we need to first ignore the existence of barriers. It means that we just use R and SD measured in expiration time instead of small intervals as what we did in explicit method. Then, we multiply the probability (my prob function) to adjust the real payoff (i.e. option price) using the given closed-form formula.

Output (I used 100000 since using 1000000 cost too much time):

```
zhujxdeMacBook-Pro:desktop zhujx$ ./continuous_barrier_option 1 0.05 0.25 50 40 100000 1000 20
--------------------------------------
European Down-and-out Continuous Barrier Options Pricing via Monte Carlo Simulation
Expiration time (Years) = 1
Risk Free Interest Rate = 0.05
Volatility (%age of stock value) = 25
Initial Stock Price = 50
Strike price = 40
Barrier price = 20
Number of Trials = 100000
Number of Divisions = 1000
--------------------------------------
--------------------------------------
The average Call Price via explicit simulation = 12.7038
The call price using (1-p)-adjusted term       = 12.7019
The theoratical Call price                     = 12.7063
--------------------------------------
The average Put Price via explicit simulation  = 0.755054
The Put price using (1-p)-adjusted term        = 0.748726
The Theoratical Put price                      = 0.751885
--------------------------------------
```

Part 2:

I would simply discuss my attempt at the Brownian bridge correction.

This time the prob adjustment is not as simple as something that can be directly calculated at T. We need to update by consecutive multiplication in our loop over our number of divisions, while a large portion of part 2 is the same as part 1.

Output:

```
[zhujxdeMacBook-Pro:desktop zhujx$ ./discrete_barrier_option 1 0.05 0.25 50 40 2000000 25 20
 --------------------------------------
 European Down-and-out Discrete Barrier Options Pricing via Monte Carlo Simulation
 Expiration time (Years) = 1
 Risk Free Interest Rate = 0.05
 Volatility (%age of stock value) = 25
 Initial Stock Price = 50
 Strike price = 40
 Barrier price = 20
 Number of Trials = 2000000
 Number of Discrete Barriers = 25
 --------------------------------------
 The average Call Price via explicit simulation of price paths            = 12.7067
 The average Call Price with Brownian-Bridge correction on the final price = 12.7075
 The average Put Price via explicit simulation of price paths             = 0.755569
 The average Put Price with Brownian-Bridge correction on the final price  = 0.754096
 --------------------------------------
```