This assignment is due at the **start** of your lecture on Tuesday, 29 October 2019.

This assignment asks you to write a few MatLab programs. Hand-in the programs and their output as well as any written answers requested in the assignment. Your programs and their output, as well as your written answers, will be marked. Your programs should be "well-commented" and use good programming style, etc. Try to format the output from your programs so that it is easy for your TA to read and to understand your results.

1. Use

   (a) simple Monte Carlo

   (b) Monte Carlo with *stratified sampling* and *Brownian Bridge*

   to price at time $t = 0$ a European *up-and-out* call option on a stock with price S_t at time $t \in [0, T]$. Assume S_t satisfies the SDE

   $$dS_t = rS_t dt + \sigma S_t dW_t$$

   in the risk-neutral world, where r is the risk-free interest rate and σ is the volatility. The parameters for the option are:

   - the initial stock price is $S_0 = \$100.00$,
   - the strike price is $K = \$100.00$ (i.e., the option is *at the money*),
   - the *up-and-out* barrier is at $B = \$110.00$,
   - the time to maturity is $T = 0.25$,
   - the risk-free interest rate is $r = 0.02$, and
   - the volatility is $\sigma = 0.2$.

   Your price should be accurate to within $\pm\$0.01$ with a confidence level of 99%.

   (a) If you use N Monte Carlo replications and discretize the interval $[0, T]$ into M subintervals of equal length, $[t_{i-1}, t_i]$, $i = 1, 2, \ldots, M$, with $t_0 = 0$ and $t_M = T$ (so that you can calculate if the Barrier is breached at t_i), how should use choose M and N for an efficient, accurate computation?

   First assume that the discretization error associated with breaching the barrier is proportional to $1/\sqrt{M}$.

   How would your answer change if the discretization error associated with breaching the barrier is proportional to $1/M$?

   Justify your answer.

   (b) Do a few test runs to see if the discretization error looks more like $1/\sqrt{M}$ or $1/M$. State your conclusion and justify your answer.

(c) Do a pilot computation to determine N so that your Monte Carlo approximation will meet the accuracy requirements.

[Hint: it might be reasonable here to assume that, if you choose M and N appropriately, then about half of the error will be due to the discretization error and half due to the Monte Carlo error. So, it might be appropriate to choose N so that the Monte Carlo error is about $\pm\$0.005$ with confidence level of 99%.

(d) How does the work needed for the two Monte Carlo methods for this assignment compare?

How does the error and computational work for this problem compare to the error and computational work for the problems in the assignments you completed earlier in the term?

[You can find a closed form solution for a European up-and-out call option in equation (7.3.19) on page 307 of Shreve's book, *Stochastic Calculus for Finance II*.]

2. Use

   (a) simple Monte Carlo, and

   (b) Monte Carlo with *importance sampling*

to price at time $t = 0$ a European *deep-out-of-the-money* call option on a stock with price S_t at time $t \in [0, T]$. Assume S_t satisfies the SDE

$$dS_t = rS_t dt + \sigma S_t dW_t$$

in the risk-neutral world, where r is the risk-free interest rate and σ is the volatility. The parameters for the option are:

- the initial stock price is $S_0 = \$80.00$,
- the strike price is $K = \$100.00$ (i.e., the option is *deep out-of-the-money*),
- the time to maturity is $T = 0.25$,
- the risk-free interest rate is $r = 0.02$, and
- the volatility is $\sigma = 0.2$.

Your price should be accurate to within $\pm\$0.0001$ with a confidence level of 99%.

Note the change in the requested accuracy compared to our earlier problems. This change is necessary because the price of this option is much lower than the price of the options that we considered in our earlier problems.

For Monte Carlo with *importance sampling*, change the drift term of the associated SDE (the way we discussed in class) so that many more of the paths will terminate *in-the-money*. Of course, you also have to incorporate the likelihood ratio, $f(x)/g(x)$, into the expectation to compensate for the higher drift, as we discussed in class.

Do a pilot computation for each Monte Carlo simulation to determine how large the number of replications, N, should be so that you price the option to the requested accuracy.

Compare your estimated prices to the Black-Scholes-formula price computed by the MatLab function `blsprice`.

How does the work for the two Monte Carlo methods compare?

Addendum: if the time to run simple Monte Carlo for this problem is too large, reduce the accuracy requested, but try to estimate how much work simple Monte Carlo would take if you had run it at the requested accuracy so that you can compare the two Monte Carlo methods.