MTH 4135 Homework 2, Due 3/2/2017

I have posted code (`StockPrice.cpp`) that simulates a stock price path over an interval of time $[0, T]$. Specifically, divide the time interval into N subintervals of length $\Delta t = \frac{T}{N}$ and put $t_i = i\Delta t$. The code takes $T = 0.5$ (years) and $N = 50$. Let $W_0, W_1, \ldots, W_N$ denote a random walk with $W_0 = 0$ and, for $i \geq 1$,

$$W_i = W_{i-1} + \begin{cases} +1 & \text{with probability } \frac{1}{2} \\ -1 & \text{with probability } \frac{1}{2}, \end{cases}$$

where the ± 1 increments are independent. Then put

$$S_{t_i} = S_0 e^{\mu t_i + \sigma\sqrt{\Delta t}\cdot W_i},$$

where $\mu = r - \frac{\ln\cosh(\sigma\sqrt{\Delta t})}{\Delta t}$. (The `cosh` function is in the `math.h` library.) Here r is the risk-free interest rate (taken to be 0.05), S_0 is today's stock price (taken to be 100), and σ is the stock price volatility (taken to be 0.30). The program uses simulation to estimate $e^{-rT} ES_T = E[e^{-rT}S_T] \approx \overline{V}$, where $\overline{V}$ is the sample mean of the simulated values of $V = e^{-rT}S_T$. The code estimates this to the nearest half-penny (`error` is 0.005) with 95% confidence and shows progress every 100000 simulations. Maintain these parameter choices throughout the following problems.

1. Run the program to verify that $\overline{V}$ agrees with S_0. (This illustrates that μ is correct.)

2. Use the fact that the random walk $\widetilde{W}_i = -W_i$ has the same distribution as W_i to generate an antithetic stock price path $\widetilde{S}_t$ with each simulation. Then estimate $e^{-rT}ES_T$ using the statistic $e^{-rT} \cdot \frac{S_T + \widetilde{S}_T}{2}$ in place of V. This should improve run-time substantially.

3. A call option on S_T struck at K has payoff at time T given by $C_T = \max(S_T - K, 0)$. Keeping the antithetic variance reduction in place, modify the code you used for problem 2 to value the call option by estimating $e^{-rT}EC_T$. That is, use the sample mean of the statistic $C^* = e^{-rT} \cdot \frac{C_T + \widetilde{C}_T}{2}$ where $\widetilde{C}_T = \max(\widetilde{S}_T - K, 0)$. Take $K = 110$.

4. Look up the Black-Scholes call option pricing formula and compute the "exact" value of this option. You will need to compute $\Psi(x) = P[\text{Normal}\,(0,1) \leq x]$ to do this. This is given by the function `Psi(x)` located in the function library. Your answers to 3 and 4 should be close, but not exact.

5. Overlay a control variable on your code for problem 3 (keeping the antithetic reduction in place) as follows. Since $EW_N = 0$, $E[W_N^2] = \operatorname{Var} W_N = N$ (confirm this), so $A = W_N^2 - N$ has mean 0. Put $C^{**} = C^* + aA$ for the appropriately chosen value of a and estimate $e^{-rT}EC_T$ using the sample mean of the C^{**} statistic. This should further reduce the run-time for valuing the option.

Over $\longrightarrow$

6. Re-run your code for problem 5 with $N = 1000$ (this will obviously take longer) to see if the estimated value approaches the exact value.

7. A "look-back" option works as follows. Let $S_{\max}$ be the maximum stock price observed along the stock price path:

$$S_{\max} \;=\; \max(S_{t_0}, S_{t_1}, S_{t_2}, \ldots, S_{t_N}).$$

($S_{t_0} = S_0 = 100$, and $S_{t_N} = S_T$.) The payoff of the look-back option at time T, L_T, is given by $L_T = \max(S_{\max} - K, 0)$. Modify the code for problem 5 (keeping all variance reduction in place and with $N = 50$ again) to value the look-back option with a strike of $K = 110$.

Extra Credit. (At my discretion, the homework group that submits the best answer to this will be awarded extra points.) The "shout" option works as follows. At any time τ between now and expiration ($0 \le \tau \le T$) the option holder can "shout" thereby locking in the price S_τ. The payoff of the option at time T is then $\max(S_T - K, S_\tau - K, 0)$. The holder can shout only once — the key is deciding when. In deciding when to shout the holder obviously has no knowledge of future stock prices, i.e., of prices S_t for $t > \tau$. Modify your code to estimate the value of this option. This is a harder problem — we will re-visit it later in the semester.

Instructions. For each problem please submit a hard copy of your code, which should be generously commented. Include screen shots showing the output. Some of your output will "run off the screen" — that's OK, just show the tail end of the output. Write an executive summary and a description of your methodology and observations/conclusions including a discussion of what the variance reduction techniques are doing and why you think they work so well.