Worcester Polytechnic Institute
Department of Mathematical Sciences
Professor: Stephan Sturm
Teaching Assistant: Qingyun Ren

Spring 2017

MA 573

Computational Methods of Mathematical Finance

Assignment 1

due on Thursday, January 26, in class

1. Use a number generator to produce samples of

   a) 10
   b) 100
   c) 1000
   d) 10000

   standard uniformly distributed random variables. For every sample size, produce a plot that shows the empirical cumulatice distrbution function $\hat{F}$ in comparison with the actual cumulative distibution function F.

2. Use a number generator to produce samples of

   a) 10
   b) 100
   c) 1000
   d) 10000

   standard uniformly distributed random variables. For every sample size, produce a histogram that shows the relative frequency of values taken, rounding to one decimal place (i.e., you count the relative frequency of realizations in bins of size 0.1).

3. A good way to check if a sequence of random variables is close to being indeed random is to make a correlation plot: Plot the sequence of pairs (x_n, x_{n+1}) generated by the random number generator inside the unit square. As less structure is visible in the plot, as better the random number generator is. Make correlation plots for a sequence of pseudorandom variables for

   a) The built-in uniform random number generator with seed $\tilde{x}_0 = 375$;
   b) A linear congruence random number generator with $m = 11$, $a = 6$ and $c = 0$ (and seed $\tilde{x}_0 = 1$);
   c) A linear congruence random number generator with $m = 2^{31} - 1$, $a = 16807$ and $c = 0$ (and seed $\tilde{x}_0 = 1$);
   d) A linear congruence random number generator with $m = 2^{31} - 1$, $a = 950706376$ and $c = 0$ (and seed $\tilde{x}_0 = 1$).

4. The generation of random number generators before the *Mersenne Twister* improved upon the basic linear congruence generator by coming different random number generators. E.g., The *Wichmann-Hill* generator implemented in Python before version 2.3 sums up over different LCRNGs and takes the fractional part of the sum. Specifically, assuming that there are K random number generators, working for $k \in \{1, \ldots, K\}$ by

$$\begin{aligned} x_{0,k} &= \tilde{x}_{0,k} \\ x_{i+1,k} &= a_k x_{i,k} \mod m_k \\ u_{i,k} &= \frac{x_{i,k}}{m_k} \end{aligned}$$

   one calculates

$$u_i = \sum_{k=1}^{K} u_{i,k} - \left\lfloor \sum_{k=1}^{K} u_{i,k} \right\rfloor$$

   (where $\lfloor x \rfloor$ denotes the largest integer smaller or equal than x).

   Consider the specific case of two LCRNGs with $\tilde{x}_{0,1} = 3, a_1 = 5, m_1 = 7$ and $\tilde{x}_{0,2} = 1, a_2 = 7, m_2 = 5$.

   a) Calculate the period length of the two LCRNGs as well as the combined Wichmann-Hill generator.
   b) Make plots for the serial correlation of all three generators (as in problem 3).

Note: All programming problems should be either in Python 2.7 (recommended) or Python 3.5, matlab, or R (no support for these languages provided). Please comment the programs *extensively* and send them in a .zip file with title **Lastname**_HW1.zip and suject line "MA 573 HW1 **Lastname**" to Qingyun Ren qren@wpi.edu before the due date of the homework (replacing the bold words by your actual last name). Plots can be provided either as printout or as .pdf file.

6 points per problems