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

Spring 2017

MA 573

Computational Methods of Mathematical Finance

Assignment 2

due on Thursday, February 2, in class

1. Use the inverse transform method to produce a standard Cauchy distributed random variable X with density
$$f_X(x) = \frac{1}{\pi(1+x^2)}$$
from a standard uniformly distributed one, $U \sim \mathcal{U}\big([0,1)\big)$:
   a) Calculate the (generalized) inverse cdf F_X^{-1} of X.
   b) Sample the Cauchy distribution numerically by creating a histogram for $x \in [-5,5)$ and bin size $\frac{1}{10}$ with 10,000 samples and compare it to a direct plot of the density f.

2. Show that the Box-Muller algorithm
$$X^1 = \cos(2\pi U^1)\sqrt{-2\log U^2}$$
$$X^2 = \sin(2\pi U^1)\sqrt{-2\log U^2}$$
produces indeed two independent standard normal random variables from independent uniform ones.

3. Use the acceptance rejection method to produce a sample of a folded normal random variable X with density

$$f_X(x) = \begin{cases} \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} & \text{if } x > 0; \\ 0 & \text{else,} \end{cases}$$

by using an exponential density as dominating term.

   a) Choose a reasonably small constant c such that $cg(x) \geq f(x)$ for all $x \in \mathbb{R}$.
   b) Plot the density of your trial by plotting a histogram for $x \in [0, 3)$ and bin size $\frac{1}{10}$ with 10,000 samples and compare it to a direct plot of the density f.

4. The *Marsaglia–Bray* method is an alternative method to generate a normal distribution out of a uniform sample, using the acceptance-rejection method: Using two independent standard normal distributed random variables, U^1, $U^2 \sim \mathcal{U}\big([0,1)\big)$, we set $V^1 = 2U^1 - 1$ and $V^2 = 2U^2 - 1$, and calculate $S = \big(V^1\big)^2 + \big(V^2\big)^2$. If $S \geq 1$ we reject the sample, otherwise we calculate

$$X^1 = V^1 \sqrt{\frac{-2\log S}{S}};$$
$$X^2 = V^2 \sqrt{\frac{-2\log S}{S}}.$$

   Prove that X^1 and X^2 are standard-normal distributed random variables.

   Note: The Marsaglia–Bray algorithm is superior to Box-Muller as it uses only one computationally expensive transcendental functions (namely log) instead of three (log, sin and cos), at the price of having a part of the sample rejected.

   Bonus question: Which percentage of the sample gets rejected?

5. Compare the Box–Muller and the Marsaglia–Bray algorithm (see problem 2 and 4) by using them to produce samples of standard-normal distributed random variables from uniform random variables given by the built-in random number generator of your programming language. Compare the densities of your samples generated by the two methods with each other and the exact normal density function by plotting a histogram for $x \in [-3, 3)$ and bin size $\frac{1}{10}$. Choose the sample size in a way that the runtime of the slower algorithm is between 1s and 10s (just using trial and error) and compare the runtimes.

6. Assume a financial markets where stock and money market account follow the dynamics

$$\begin{aligned} dS_t &= \mu S_t\,dt + \sigma S_t\,dW_t, && S_0 = 100; \\ dB_t &= rB_t\,dt, && B_0 = 1, \end{aligned}$$

with $\mu = 0.05$, $\sigma = 0.2$ and $r = 0.03$.

(a) Calculate the price of a European put option with maturity $T = 0.5$ and strike $K = 110$ using the Black–Scholes formula.

(b) Calculate the price of a European put option with maturity $T = 0.5$ and strike $K = 110$ using Monte-Carlo integration.

(c) Compare the results of part (a) and part (b), using different sample sizes for the Monte-Carlo integration. Make a plot that shows the numerically computed option price as function of the sample size.

(d) Calculate the price of a European asset-or-nothig digital call option (payoff $S_T \mathbb{1}_{\{S_T > K\}}$) option with maturity $T = 0.5$ and strike $K = 110$ using Monte-Carlo integration.

(e) Calculate the price of a European cubic put option (payoff $\left((K - S_T)^+\right)^3$) option with maturity $T = 0.5$ and strike $K = 110$ using Monte-Carlo integration.

(f) Calculate the price of a European gap call option (payoff $(S_T - L)^+ \mathbb{1}_{\{S_T > K\}}$) option with maturity $T = 0.5$ and strike $K = 110$ and exercise level $L = 105$ using Monte-Carlo integration.

(g) Calculate the price of a European exponential put option (payoff $e^{(K-S_T)^+}$) option with maturity $T = 0.5$ and strike $K = 110$ using Monte-Carlo integration.

Bonus question: Which of the options of problem (d)–(g) admit a closed-form representation of the option price?

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**_HW2.zip and suject line "MA 573 HW2 **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.

4 points per problems