Homework Monte Carlo methods

1. <u>Monte Carlo methods</u>

   Consider the ratio of integrals

$$r_m = \frac{\displaystyle\int_{-\infty}^{\infty} x^m \cdot \frac{1}{b\pi \cdot \left[1 + \left(\frac{x-a}{b}\right)^2\right]} \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(x-\mu)^2/2\sigma^2} \; dx}{\displaystyle\int_{-\infty}^{\infty} \frac{1}{b\pi \cdot \left[1 + \left(\frac{x-a}{b}\right)^2\right]} \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(x-\mu)^2/2\sigma^2} \; dx}$$

   which can be seen as the expected value $r_m = E(X^m)$, where

$$f_X(x) = \frac{\frac{1}{b\pi \cdot \left[1 + \left(\frac{x-a}{b}\right)^2\right]} \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(x-\mu)^2/2\sigma^2}}{I}$$

   and

$$I = \int_{-\infty}^{\infty} \frac{1}{b\pi \cdot \left[1 + \left(\frac{x-a}{b}\right)^2\right]} \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(x-\mu)^2/2\sigma^2} \; dx$$

   Consider for instance the case $m = 1$. The expected value can be computed with three methods. In this question, implement all three and compare the accuracy of the three methods. The parameters a, b, μ and σ are free to choose.

   (a) Use **rejection sampling** to generate data X from $f_X(x)$ and estimate $\theta = r_1 = EX$ by $\hat{\theta}_1 = \overline{X}$. For this, you can use the matlab routine `randCauchyplusNormal.m`.

   (b) Generate normal data $Z \sim N(\mu, \sigma^2)$ and estimate EX by $\hat{\theta}_2 = \dfrac{\overline{Z \cdot I_Z}}{\overline{I_Z}}$ with

$$I_Z = \frac{1}{b\pi \left[1 + \left(\frac{Z-a}{b}\right)^2\right]}$$

   (c) Generate Cauchy data $T \sim \text{Cauchy}(a, b)$ and estimate EX by $\hat{\theta}_3 = \dfrac{\overline{T \cdot I_T}}{\overline{I_T}}$ with

$$I_T = \exp\left[-(T-\mu)^2/2\sigma^2\right]/\sqrt{2\pi}\sigma$$

   Repeat each estimation procedure to be able to estimate its accuracy. Perform a statistical test (ANOVA-type, for instance Levene's test) to check whether $\text{var}(\hat{\theta}_1) = \text{var}(\hat{\theta}_2) = \text{var}(\hat{\theta}_3)$. Briefly discuss the bias of each of the three samplers.

2. Markov Chain Monte Carlo sampling

The matlab file `randgammaMCMC.m` implements a Metropolis-Hastings algorithm for the generation of pseudo random variables following a Gamma distribution. It is an alternative to the routine `randgamma.m`, which uses rejection sampling. The file `testrandgammaMCMC.m` compares both generators.

(a) In order to generate $X \sim \Gamma(\lambda, \alpha)$, meaning that

$$f_X(x) = \frac{x^{\alpha-1}\lambda^{\alpha}e^{-\lambda x}}{\Gamma(\alpha)}$$

(as on slide 40), the Metropolis-Hastings algorithm uses a proposal distribution

$$q(x|y) = \frac{x^{r-1}\lambda^{r}e^{-\lambda x}}{(r-1)!},$$

where $r = \lfloor \alpha \rfloor$ is the floor function (i.e., the greatest integer less than or equal to) of α. In other words, the proposal distribution is an Erlang distribution (= Gamma with integer parameter = distribution of the sum of independent, identically distributed exponential variables). Note that the proposal distribution does not depend on the current value (i.e., $q(x|y)$ does not depend on y).

Knowing this, explain how to arrive at the lines of code in `randgammaMCMC.m`, more precisely the lines

```
Y = randexp(1,r,lambda); X(k) = sum(Y);
a = (X(k)/X(k-1))^ralfa;
U = rand;
if U<a, reject = false; end
```

(b) Run the file `testrandgammaMCMC.m` and explain in one or two sentences what you see. What could be a drawback of `randgammaMCMC.m` (MCMC sampling) compared to `randgamma.m` (rejection sampling).