MMF 2021 **Midterm Test** 8 November 2019

This is a **closed-book test**: no books, no notes, no calculators, no phones, no tablets, no computers (of any kind) allowed.

Do **NOT** turn this page over until you are **TOLD** to start.

Duration of the test: 3 hours
(although I think you should be able to do it in much less than 3 hours).

Write your answers in the test booklets provided.

Please fill-in **ALL** the information requested on the front cover of **EACH** test booklet that you use.

The test consists of 6 pages, including this one. Make sure you have all 6 pages.

The test consists of 4 questions. **Answer all 4 questions**. The mark for each question is listed at the start of the question.

The test was written with the intention that you would have ample time to complete it. You will be rewarded for concise well-thought-out answers, rather than long rambling ones. **We seek quality rather than quantity**.

Moreover, an answer that contains relevant and correct information as well as irrelevant or incorrect information will be awarded fewer marks than one that contains the same relevant and correct information only.

Write legibly. Unreadable answers are worthless.

1. [10 marks: 5 marks for each part]

   Consider the expression

$$\sqrt{1+x} - \sqrt{1-x} \tag{1}$$

   for $x \in [-1, 1]$.

   (a) For what range of values of x does expression (1) produce inaccurate results in IEEE double-precision floating-point arithmetic?
   (By inaccurate I mean that the absolute value of the relative error in the floating-point approximation to (1) is orders of magnitude larger than *machine epsilon* for this floating-point number system.)

   Justify your answer.

   (b) Give another expression that is mathematically equal to (1), but is computationally much more effective than (1) in the sense that it is accurate in IEEE double-precision floating-point arithmetic for all $x \in [-1, 1]$.
   (By accurate I mean that the the absolute value of the relative error associated with your new floating-point expression is less than 10 times *machine epsilon*.)

   Justify your answer.

2. [5 marks]

   In the course lectures, we discussed the acceptance-rejection method for continuous random varaibles. However, you can extend the acceptance-rejection method to discrete random variables as follows.

   Suppose X is a discrete random variable with probability mass function (pmf)

$$\mathbb{P}(X = k) = p(k) \qquad \text{for } k = 0, 1, 2, \dots$$

   Assume there is another discrete random variable Y with pmf

$$\mathbb{P}(Y = k) = q(k) \qquad \text{for } k = 0, 1, 2, \dots$$

   In addition, assume you can generate a Y with pmf q and that there is a constant $c < \infty$ such that

$$p(k) \le c\, q(k) \qquad \text{for } k = 0, 1, 2, \dots$$

   The acceptance-rejection method to generate a discrete random variable X with pmf p is as follows.

```
1. Generate a discrete random variable $Y$ with pmf $q$.
2. Generate $U \sim \mathrm{Unif}[0,1]$ independent from $Y$.
3. If $U \le \frac{p(Y)}{c\,q(Y)}$ then
       Set $X = Y$ and return $X$
   else
       Go to step 1 and repeat the procedure
   end if
```

   Prove that the random variable X returned by the algorithm above has pmf p.
   That is, prove $\mathbb{P}(X = k) = p(k)$ for $k = 0, 1, 2, \dots$, where X is the random variable returned by the algorithm above.

3. [5 marks]

When we were discussing importance sampling in class, one of the examples we considered was to approximate the integral

$$\int_0^1 h(x)\,dx$$

by both simple Monte Carlo and Monte Carlo with importance sampling. In the particular example we considered in class, $h(x) = \sqrt{1-x^2}$. Our goal was to find a probability density function (pdf), $g(x)$, such that

$$\frac{h(x)}{g(x)} \approx c$$

for some constant c. We accomplished this goal by

- dividing the interval $[0,1]$ into L subintervals, $[(k-1)/L, k/L]$, for $k = 1, 2, \ldots, L$,
- letting $s_k = (k-1)/L + 1/(2L)$ be the midpoint of the k^{th} subinterval, $[(k-1)/L, k/L]$,
- setting

  $$q_k = \frac{h(s_k)}{\sum_{j=1}^{L} h(s_j)}$$

  for $k = 1, 2, \ldots, L$,
- letting

  $$g(x) = Lq_k \qquad \text{for } x \in [(k-1)/L, k/L)$$

  and $g(1) = Lq_L$.

Note that, for $h(x) = \sqrt{1-x^2}$, $h(s_k) > 0$ for all $k = 1, 2, \ldots, L$. Hence, $q_k > 0$ for all $k = 1, 2, \ldots, L$. Moreover,

$$\sum_{k=1}^{L} q_k = 1$$

So, we can view q_k as a probability. Furthermore, for $x \in [(k-1)/L, k/L)$,

$$\begin{aligned} \frac{h(x)}{g(x)} &= \frac{h(x)}{Lq_k} \\ &= \frac{\sum_{j=1}^{L} h(s_j)}{L} \, \frac{h(x)}{h(s_k)} \\ &= c\,\frac{h(x)}{h(s_k)} \\ &\approx c \end{aligned} \tag{2}$$

since $h(x) \approx h(s_k)$ for $x \in [(k-1)/L, k/L)$. Note that we used

$$c = \frac{\sum_{j=1}^{L} h(s_j)}{L}$$

in (2) above.

Also, note that $g(x) > 0$, since, as noted above, $q_k > 0$ for all $k = 1, 2, \ldots, L$. Moreover,

$$\begin{aligned}\int_0^1 g(x)\,dx &= \sum_{k=1}^{L} \int_{(k-1)/L}^{k/L} g(x)\,dx \\ &= \sum_{k=1}^{L} \int_{(k-1)/L}^{k/L} Lq_k\,dx \\ &= \sum_{k=1}^{L} Lq_k\,\frac{1}{L} \\ &= \sum_{k=1}^{L} q_k \\ &= 1\end{aligned}$$

Hence, $g(x)$ is a pdf.

To implement importance sampling for this problem, we need to generate random variables, $X_1, X_2, X_3, \ldots,$ each with probability distribution defined by the pdf $g(x)$. In class, I gave you one way to generate such a random variable, X. This method works as follows.

- Generate a $U_1 \sim \text{Unif}[0,1]$
- Find the $k \in \{1, 2, \ldots, L\}$ such that

  $$\sum_{j=1}^{k-1} q_k \le U_1 < \sum_{j=1}^{k} q_k$$

  where we take

  $$\sum_{j=1}^{0} q_k = 0$$

- Generate a $U_2 \sim \text{Unif}[0,1]$
- Set

  $$X = \frac{k-1}{L} + \frac{U_2}{L}$$

The method above requires two $\text{Unif}[0,1]$ random variables for each X that it generates.

Describe another method for generating a random variable, X, with probability distribution defined by $g(x)$, that requires only one $\text{Unif}[0,1]$ random variable for each X that it generates.

4. [15 marks: 5 marks for each part]

In many applications, you need to compute tail probabilities, $\mathbb{P}(X \geq x)$. For example, if $\mathcal{L}$ is the loss associated with defaults in a portfolio of bonds over some time period T, you might want to compute $\mathbb{P}(\mathcal{L} \geq l)$. This is required, for example, if you want to compute the Value-at-Risk (VaR) associated with the portfolio.

However, to keep things simple in this question, we'll focus on computing $\mathbb{P}(X \geq 10)$, where $X \sim N(0,1)$ is a standard normal random variable (i.e., X is a normal random variable with mean 0 and variance 1). You might think that you can compute $\mathbb{P}(X \geq 10)$ from the CDF, $\Phi(x)$, of the normal distribution, since $\mathbb{P}(X \geq 10) = 1 - \Phi(10)$. Although this is true in theory, if you evaluate this in MatLab, you'll find that the computed value of $\Phi(10)$ is 1, whence the computed value of $1 - \Phi(10)$ is 0. So, this does not lead to a good approximation to $\mathbb{P}(X \geq 10)$. The reason for this is that $\mathbb{P}(X \geq 10) \approx 10^{-23}$. So, your computer would need to carry the equivalent of at least 23 decimal digits to be able to differentiate $\Phi(10)$ from 1.

Even if we could compute $\Phi(10)$ to sufficient accuracy, this would not help in the more realistic example $\mathbb{P}(\mathcal{L} \geq l)$ mentioned above. So, let's forget about computing $\mathbb{P}(X \geq 10)$ from $\Phi(x)$ for now.

Another approach is to approximate $\mathbb{P}(X \geq 10)$ by a Monte Carlo simulation. To this end, note that

$$\mathbb{P}(X \geq 10) = \int_{10}^{\infty} f(x)\,dx = \int_{-\infty}^{\infty} H_{10}(x) f(x)\,dx = \mathbb{E}_f[H_{10}(X)]$$

where $X \sim N(0,1)$,

$$f(x) = \frac{\mathrm{e}^{-x^2/2}}{\sqrt{2\pi}}$$

and

$$H_{10}(x) = \begin{cases} 1 & \text{if } x \geq 10 \\ 0 & \text{if } x < 10 \end{cases}$$

Hence, we can write a very simple Monte Carlo simulation to approximate
$p = \mathbb{P}(X \geq 10) = \mathbb{E}_f[H_{10}(X)]$:

$$\hat{p} = \frac{1}{N} \sum_{i=1}^{N} H_{10}(X_i) \tag{3}$$

where $X_i \sim N(0,1)$. Suppose we want $\hat{p}$ to approximate $p = \mathbb{P}(X \geq 10)$ to at least two significance digits with a 95% confidence level.

(a) First show that the variance of $H_{10}(X)$ satisfies

$$\mathrm{Var}_f[H_{10}(X)] = \mathbb{E}_f[(H_{10}(X) - p)^2] = p - p^2 \tag{4}$$

Then use the value of $\mathrm{Var}_f[H_{10}(X)]$ given by (4) (even if you were not able to verify (4)) to estimate how large you need to choose N in (3) to achieve this level of accuracy.

Your estimation of N needs to be of the right order of magnitude only. So, you can use $z_{\delta/2} \approx 2$ for the 95% confidence level and $p \approx 10^{-23}$ in computing your estimate of N.

Your value of N computed in part (a) above should be so large that the Monte Carlo simulation (3) is completely impractical. However, we can use importance sampling to get a much more efficient Monte Carlo simulation. To this end, let

$$g(x) = \frac{\mathrm{e}^{-(x-10)^2/2}}{\sqrt{2\pi}}$$

be the probability density function for $Y \sim N(10,1)$. Then

$$\begin{aligned} p = \mathbb{P}(X \geq 10) &= \int_{10}^{\infty} f(x)\,dx \\ &= \int_{-\infty}^{\infty} H_{10}(x) f(x)\,dx \\ &= \int_{-\infty}^{\infty} H_{10}(x) \frac{f(x)}{g(x)} g(x)\,dx \\ &= \mathbb{E}_g\left[H_{10}(Y)\frac{f(Y)}{g(Y)}\right] \end{aligned} \tag{5}$$

where $X \sim N(0,1)$ and $Y \sim N(10,1)$.

(b) What is the Monte-Carlo importance-sampling simulation associated with $\mathbb{E}_g[H_{10}(Y)\frac{f(Y)}{g(Y)}]$ in (5) above to approximate $p = \mathbb{P}(X \geq 10)$?

Clearly state what random variables you are using in this Monte Carlo simulation and how you would compute them if you have a function, such as MatLab's randn, that returns $N(0,1)$ normal random variables.

Let

$$\mathrm{Var}_g\left[H_{10}(Y)\frac{f(Y)}{g(Y)}\right] = \mathbb{E}_g\left[\left(H_{10}(Y)\frac{f(Y)}{g(Y)} - p\right)^2\right]$$

where $Y \sim N(10,1)$. To assess the efficiency of the Monte-Carlo importance-sampling simulation in part (b) above compared to the simple Monte-Carlo simulation (3), we need to estimate how much smaller the variance $\mathrm{Var}_g\left[H_{10}(Y)\frac{f(Y)}{g(Y)}\right]$ is than the variance $\mathrm{Var}_f[H_{10}(X)]$.

It does not seem too easy to get a closed form expression for $\mathrm{Var}_g\left[H_{10}(Y)\frac{f(Y)}{g(Y)}\right]$, but it is not too hard to show

$$\mathrm{Var}_g\left[H_{10}(Y)\frac{f(Y)}{g(Y)}\right] \leq \mathrm{e}^{-50}p - p^2 \tag{6}$$

Thus, $\mathrm{Var}_g\left[H_{10}(Y)\frac{f(Y)}{g(Y)}\right]$ is at least approximately $\mathrm{e}^{-50} \approx 2 \times 10^{-22}$ times smaller than $\mathrm{Var}_f[H_{10}(X)]$. This is quite a significant variance reduction!

(c) Show that (6) is true.

Then use (6) (even if you were not able to prove it is true) to estimate how large you need to choose N in your Monte-Carlo importance-sampling simulation to achieve the same level of accuracy as was specified in part (a) above.

Your estimation of N needs to be of the right order of magnitude only. So, you can use $z_{\delta/2} \approx 2$ for the 95% confidence level, $p \approx 10^{-23}$ and $\mathrm{e}^{-50} \approx 2 \times 10^{-22}$ in computing your estimate of N.