Math 350 - Homework 6 - Solutions

1. *We wish to generate a random variable X taking values in $[0,1]$, having probability density function*

$$f(x) = e^x/(e-1).$$

   (a) *Write a program based on the inverse transform method to generate X.*

   (b) *Show that your program works as desired by doing the following: obtain a graph of the approximate density function using a suitably normalized histogram plot from a large number of simulated values of X; then superpose the graph of the exact function $f(x)$ to the first graph.*

The cumulative distribution function is

$$F(x) = \int_0^x f(s)\,ds = \frac{e^x - 1}{e-1}$$

for $0 \leq x \leq 1$. The function $u = F(x)$ is easily inverted: $x = G(u)$ where

$$G(u) = \ln(1 + (e-1)u).$$

The program and comparison of the simulated data with the exact probability density can be done in Matlab as follows:

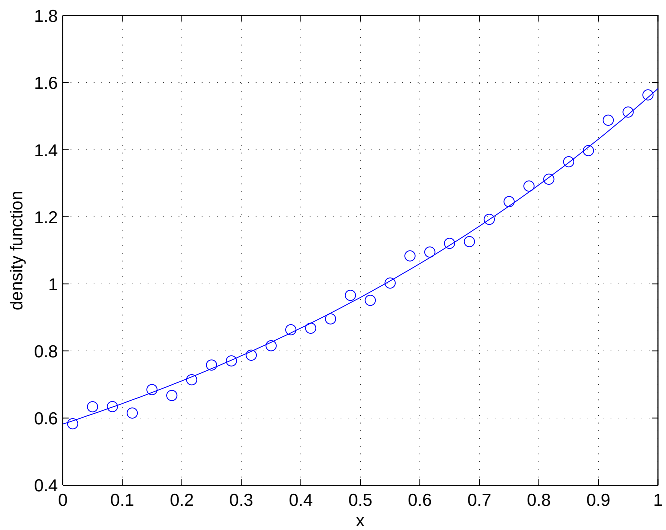
```
%The following generates m sample values of X:
m=100000;
U=rand(1,m);
X=log(1+(exp(1)-1)*U);

%A histogram plot can be obtained as follows (the number of
%bins I'm using (30) is somewhat arbitrary):
d=1/30;
[r,s]=hist(X,30);
%r gives the occupation numbers of the bins, and s the x-coordinate
%of the center of each bin.
r=r/(d*m);
%By normalizing the values this way
%the Riemann sum (r_1 + ...+ r_30)*d is one, so this r can better be compared with
%the actual density function f(x).
plot(s,r,'o')

%I have used an ordinary plot with circles marking the data points for
%better visual effect, since I now want to superpose the exact density
%function to it.
```

```
hold on
x=0:0.01:1;
y=exp(x)/(exp(1)-1);
plot(x,y)
grid
xlabel('x')
ylabel('density function')
```

The solid line in the following graph gives the exact density function $f(x) = e^x/(e-1)$, and the circles are the empirical values obtained by simulation.



2. *Suppose it is relatively easy to generate random variables from any of the distributions* F_i $i = 1, \ldots, n$. *How could we generate a random variable having the distribution*

$$F(x) = \sum_{i=1}^{n} p_i F_i(x)$$

where p_i, $i = 1, \ldots, n$, *are nonnegative numbers whose sum is* 1*?*

Let $X_1, \ldots, X_n$ be random variables having cumulative distribution functions $F_1(x), \ldots, F_n(x)$, respectively. Let I be a random variable taking values in $\{1, \ldots, n\}$ with probabilities

$$P(I = i) = p_i.$$

Now define the random variable

$$X = X_I.$$

In other words, a value of X is obtained by generating a value i of I (with probability p_i) and then independently a value of X_i using the distribution function $F_i(x)$. To see that X has the correct distribution function, note:

$$\begin{aligned} P(X \le x) &= \sum_{i=1}^{n} P(X_I \le x | I = i) P(I = i) \\ &= \sum_{i=1}^{n} P(X_i \le x) p_i \\ &= \sum_{i=1}^{n} F_i(x) p_i \end{aligned}$$

3. *Using the result of Exercise 7, give an algorithm for generating random variables from the cumulative distribution function*

$$F(x) = \frac{x + x^3 + x^5}{3}, \ 0 \le x \le 1.$$

Let $F_1(x) = x$, $F_2(x) = x^3$, and $F_3(x) = x^5$. Note that $x = F_i^{-1}(u) = u^{1/(2i-1)}$ for $i = 1, 2, 3$. With this in mind, we can describe an algorithm for generating a random variable X with distribution function $F(x)$ as follows:

(a) Generate a random number U uniformly distributed over $(0, 1)$;

(b) Let $I = \text{Int}(3 * U) + 1$; (this produces a random index from $\{1, 2, 3\}$ with probabilities 1/3.)

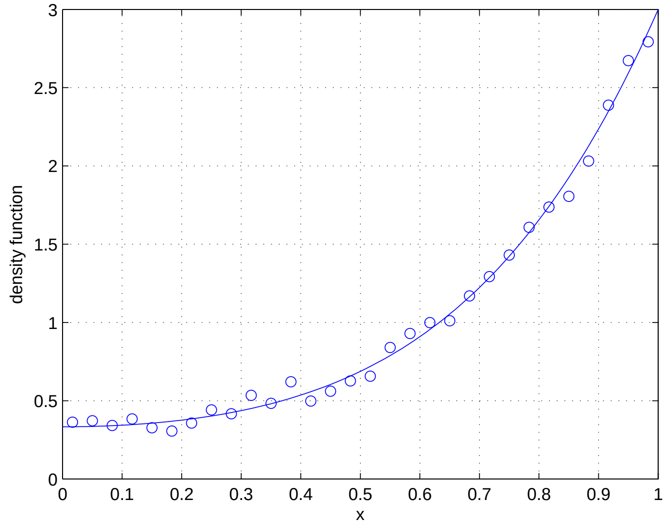
(c) Generate another random number V independent of U, also uniform over $(0, 1)$;

(d) Now set $X = V^{1/(2I-1)}$.

The following program implements this algorithm in Matlab:

```
%Generate m independent sample values of X by the composition method:
m=10000;
U=rand(1,m);
I=floor(3*U)+1;
V=rand(1,m);
X=V.^(1./(2*I-1));

%We can compare the approximate density obtained by simulation
%and the exact one, which is f(x)=F'(x)=(1+3x^2+5x^4)/3. We do this
%as in problem 1.
d=1/30;
[r,s]=hist(X,30);
r=r/(d*m);
plot(s,r,'o')
hold on
x=0:0.01:1;
y=(1+3*x.^2+5*x.^4)/3;
plot(x,y)
grid
xlabel('x')
ylabel('density function')
```



4. *A casualty insurance company has* 1000 *policy holders, each of whom will independently present a claim in the next month with probability* 0.05*. Assuming that the amounts of the claims made are independent exponential random variables with mean* \$800*, use simulation to estimate the probability that the sum of these claims exceeds* \$50, 000.

The number N of claims in the next month is a binomial random variable with parameters $n = 1000$ and $p = 0.05$. And the value of each individual claim is an independent exponential random variable with parameter $\lambda = 1/800$.

To generate the number of claims we may in the present situation (since n is big and p is small) use the Poisson approximation of the binomial distribution. (See Section 2.8, page 20 of the textbook.) In other words, we set $\mu = np = 50$ and write

$$P(N = i) = e^{-\mu}\frac{\mu^i}{i!}.$$

I will use the following algorithm to simulate the sum S of all claims:

(a) Generate a Poisson random variable N with mean $\mu = 50$;

(b) Generate i.i.d. exponential random variables $S_1, \dots, S_N$ having parameter $\lambda = 1/800$;

(c) Let $S = S_1 + \cdots + S_N$.

To simulate the Poisson random variable I use the algorithm on page 56 of the text. In Matlab, the program can be written as follows:

```
function X=Poisson_rv(lambda)
%Input:  the mean value lambda of the Poisson random variable;
%Output: one sample value of the Poisson random variable.
%
%Uses algorithm of textbook, page 56, based on the inverse transformation
%method.
i=0;
p=exp(-lambda);
F=p;
```

```
U=rand(1);
while U>=F
    p=lambda*p/(i+1);
    F=F+p;
    i=i+1;
end
X=i;
```

The following program now estimates the asked for probability in terms of the empirical frequency f of total claims more than $\$50,000$.

```
m=10000;%Number of trials of the experiment
mu=50;%Parameter (mean) of the Poisson random variable N.
lambda=1/800;%Parameter (rate) of the exponential r.v.
%Let S be the vector of m sample values of the total claim amount in
%dollars.
S=zeros(1,m);
for i=1:m
    N=Poisson_rv(mu);%Number of claims. Instead of using the Poisson_rv program, we
    %could simply simulate the binomial random variable N=sum(rand(1,1000)<=0.05)
    U=rand(1,N);
    X=-(1/lambda)*log(U);%Vector of claim amounts in dollars.
    S(i)=sum(X);%Total amount claimed.
end
%To estimate the probability that the total claim is more than 50,000
%dollars, we can calculate the empirical frequency by
f=sum(S>=50000)/m
```

A typical value obtained for f was: 0.1080.

In order to gain some confidence that this is a reasonable number, let us think about what kind of probability would result by using the Central Limit Theorem. Let $a = \$50,000$. The probability $P(S \geq a)$ can be written as follows:

$$
\begin{aligned}
P(S \geq S_0) &= \sum_{j=0}^{\infty} P(S_1 + \cdots + S_N \geq a | N = j) P(N = j) \\
&= \sum_{j=0}^{\infty} P(S_1 + \cdots + S_j \geq a) e^{-\mu} \frac{\mu^j}{j!} \\
&= 1 - \sum_{j=0}^{\infty} P(S_1 + \cdots + S_j < a) e^{-\mu} \frac{\mu^j}{j!}.
\end{aligned}
$$

Now, the probabilities $P(N = j)$ are concentrated in a range of values of j around the mean 50. A little experimentation shows that the probabilities for $j \leq 20$ and $j > 80$ are small enough that we can discard the corresponding terms in the sum. On the other hand, since each S_i is exponential with mean 800 and variance 800^2, we write

$$P(S_1 + \cdots + S_j < a) = P\left(\frac{S_1 + \cdots + S_j - 800j}{800\sqrt{j}} < \frac{a - 800j}{800\sqrt{j}}\right) \approx \Phi\left(\frac{a - 800j}{800\sqrt{j}}\right),$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal. So the probability we want can be approximated by

$$p \approx 1 - \sum_{j=20}^{80} \Phi\left(\frac{50000 - 800j}{800\sqrt{j}}\right) e^{-50} \frac{50^j}{j!}.$$

In Matlab the function $\Phi(x)$ is called `normcdf(x)`. The approximation for p that I've obtained using the above expression is 0.1087.

5. *Buses arrive at a sporting event according to a Poisson process with a rate* 5 *per hour. Each bus is equally likely to contain either* $20, 21, \ldots, 40$ *fans, with the numbers in the different buses being independent. Write an algorithm to simulate the arrival of fans to the event by time* $t = 1$.

An algorithm can be as follows:

(a) Generate a Poisson random variable N with parameter $\lambda = 5$;

(b) For each $i \in \{0, 1, \ldots, N\}$ let $X_i = 20 + \mathrm{Int}(21 * U_i)$ be a uniformly distributed number over the set $\{20, 21, \ldots, 40\}$, where $U_1, \ldots, U_N$ are i.i.d. uniform over $(0, 1)$;

(c) Then the total number of fans that arrive in one hour is $X = X_1 + \cdots + X_N$.

We may implement this in Matlab as follows (the program simulates m trials of X and computes the average, which we expect to be $5 \times 30 = 150$):

```
m=10000;
X=zeros(1,m);
for i=1:m
    N=Poisson_rv(5);
    U=rand(1,N);
    X(i)=sum(20*ones(1,N)+floor(21*U));
end
a=sum(X)/m
```

A typical value obtained from this program was: 149.29.