MATH 565 Group 12 Mikel Mendibe and Javier Echavarren

1.For the situation in Assignment 1 problem 1, construct a confidence interval for the probability that the time to wait of a bus is greater than 8 minutes using 10^6 Monte Carlo samples and binomialCI.

The confidence interval for the probability that the time to wait of a bus is greater than 8 minutes is [0.466238,0.467051], the code is below.

Matlab Code:

```
N=10e6
Wait_TimeHW2=8;

Times = Bus(N);
m=mean(Times) %we extract the mean to compare with the model we did in
python
successes = sum(Times>Wait_TimeHW2) %Counts the successes

Confidence_Interval= binomialCI(N,successes,0.01) %delivers the Confidence
Interval

fprintf('The confidence interval  for the probability that the time to wait
of a bus is greater than 8 minutes is [%f,%f] ' , Confidence_Interval);

function Waiting_times = Bus(Sim) %function that gives a vector of times
waited by the person, of dimension Sim.
    Arrival_time=rand(Sim,1)*15;
    bus1=-1+rand(Sim,1)*3;
    bus2=14+rand(Sim,1)*3;
    bus3=29+rand(Sim,1)*3;
    Take_bus_1=bus1>Arrival_time;
    Take_bus_2=xor(Take_bus_1,bus2>Arrival_time);
    Take_bus_3=Take_bus_1==Take_bus_2;
    Waiting_times=(bus1-Arrival_time).*Take_bus_1+(bus2-
Arrival_time).*Take_bus_2+(bus3-Arrival_time).*Take_bus_3;
    end
```

2.Consider the following three dimensional integral

$$\boldsymbol{\mu = \int_{[-1,1]^3} cos\left(\sqrt{x_1^2 + x_2^2} + x_3\right) dx.}$$

The following are six IID U[0, 1] random numbers:

0.1135, 0.9745, 0.7287, 0.3515, 0.7076, 0.7996

Use these to form $\hat{\mu}_n$, a Mont Carlo estimate of μ. Granted, n cannot be very large given only six IID U[0, 1] random numbers, how large can you n be?

To form the estimate of μ, $\hat{\mu}_n$, we firstly make a change of variable,

$x = -1 + 2y, \quad Y \sim U(0,1)$

$$\mu = \int_{[-1,1]^3} cos\left(\sqrt{x_1^2 + x_2^2} + x_3\right) dx$$

$$\mu = \int_{[0,1]^3} cos\left(\sqrt{(-1+2y_1)^2 + (-1+2y_2)^2} + -1 + 2y_3\right) \cdot \prod_{j=1}^{3} 2\, dy$$

Now that we have changed the variable correctly, we only have to substitute the values of (y_1, y_2, y_3) with the values generated randomly and make the mean. As we have six random numbers, and we want the variables entering the calculation to be IID, we can only make two trios.

Trio 1: $(y_1, y_2, y_3) = (0.1135, 0.9745, 0.7287)$

$$n_1 = cos\left(\sqrt{(-1+2\cdot 0.1135)^2 + (-1+0.9745)^2} + -1 + 2\cdot 0.7287\right) \cdot \prod_{j=1}^{3} 2\, dy$$

$$n_1 = cos\left(\sqrt{(-0.773)^2 + (0.949)^2} + 0.4574\right) \cdot 8$$

$$n_1 = -0.88288$$

Trio 2: $(y_1, y_2, y_3) = (0.3515, 0.7076, 0.7996)$

$$n_2 = cos\left(\sqrt{(-1+2\cdot 0.3515)^2 + (-1+0.7076)^2} + -1 + 2\cdot 0.7996\right) \cdot \prod_{j=1}^{3} 2\, dy$$

$$n_2 = cos\left(\sqrt{(-0.297)^2 + (0.4152)^2} + 0.5992\right) \cdot 8$$

$$n_2 = 3.55951$$

$$\hat{\mu}_2 = \frac{n_1 + n_2}{2} = 1.3383$$

Our n can be only two, because we must only use each random number once because the variables entered in the model must be IID, and if we reused numbers, they wouldn't be independent.

3.Computer Problem: Consider the following integral, which arises in the pricing of an Asian arithmetic mean call option:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \max\left(\frac{1}{2}[S_1(\boldsymbol{z}) + S_2(\boldsymbol{z})] - 100, 0\right) \frac{\exp(-\boldsymbol{z}^T\Sigma^{-1}\boldsymbol{z}/2)}{\sqrt{4\pi^2 \det(\Sigma)}} d\boldsymbol{z}, \qquad \Sigma = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1 \end{pmatrix},$$

$$S_1(\boldsymbol{z}) = 100\exp(-0.0225 + 0.3z_1), \qquad S_2(\boldsymbol{z}) = 100\exp(-0.045 + 0.3z_2).$$

Approximate the integral using meanMC g with an error tolerance of 0.02.

Following the Lecture Notes, we use affine transformation to compute the solution. We are already given $\rho(x)$, and we know that $f(x) = g(Ax)$, and

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i = \frac{1}{n}\sum_{i=1}^{n} f(X_i), \qquad X_i \sim N(0, I_2)$$

So, we only need to discover a matrix A that follows $AA^T = \sum$,

and we choose $A = \begin{pmatrix} 1/2 & 1/2 \\ 0 & 1 \end{pmatrix}$. With this A chosen, we only have to produce n=10000 random normal vectors 2x1 (X), make the transformation AX, introduce it to f so that we get $g(Ax) = f(x)$, and then compute the standard deviation.

To get a confidence interval with an error of less than 0.02, from the Central Limit Theorem, assuming an inflation factor of 1.2 and and a 99% confidence, we know that the number of samples needed are the following:

$$n_\mu = \left\lceil \left(\frac{2.58 \cdot 1.2\hat{\sigma}}{\varepsilon_a} \right)^2 \right\rceil$$

So we compute again $\hat{\mu}_n$, but taking $n_\mu = 4470600$ samples, obtaining that the expected value is $\hat{\mu}_n = 9.4215$, and the absolute error is $\varepsilon = 0.0164$, less than what was asked (because of the use of the inflation factor). The Matlab code is below.

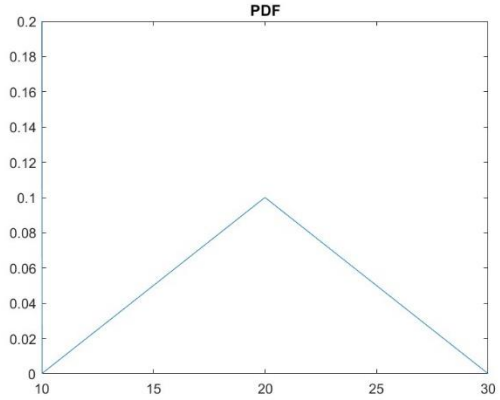
```
Ntrial3=1e4
Sample_Vector3=integral3(Ntrial3); %We compute 1e4 simulations to estimate
the standars deviation.
plot( 1:Ntrial3 ,Sample_Vector3);
mean3= mean(Sample_Vector3)


Nintegral3= round(((2.58*1.2*std(Sample_Vector3))/0.02)^2),%We calculate
the number of simulations needed
Integral_Vector3=integral3(Nintegral3);
mean3= mean(Integral_Vector3)%we get the mean of the whole simulation
Error = (2.58*std(Integral_Vector3))/sqrt(Nintegral3) %We get the error to
see if it is in the margin we wanted

function Integral_vector = integral3 (Sim) %The function that does n
simulations of the function, returns an n vector of each simulation.
    Integral_vector = zeros(Sim,1);
        A = [1/sqrt(2) 0;1/sqrt(2) 1/sqrt(2)];
        X = normrnd(0,1,2,Sim);
        Z = A*X;
        S=100*exp(-0.0225+0.3*Z(1,:))+100*exp(-0.045+0.3*Z(2,:));
        Integral_vector=max(0.5*S-100,0);
end
```

4. Y has a triangular probability density function (PDF) of

$$\rho(y)\begin{cases}\dfrac{y-10}{100} & 10 \le y < 20\\ \dfrac{30-y}{100} & 20 < y < 30\\ 0 & otherwise\end{cases}$$

In order for $\rho(y)$ to be a PDF, it needs to comply with the following characteristics:

- $\rho(y)$ always greater or equal to zero. This can be proved by looking at the representation of the Probability Density Function.



- $\rho(y)$ is continuous over the entire range. This can be proved by looking at the representation of the Probability Density Function.
- $\int_{-\infty}^{\infty} \rho(y)\, dy = 1 \rightarrow \int_{10}^{20} \frac{y-10}{100}\, dy + \int_{20}^{30} \frac{30-y}{100}\, dy = 1$

  It has been proven that $\rho(y)$ is a Probability Density Function.

Verify that ρ is a PDF. Use the inverse transform method to generate Y using the following six IID U[0, 1] random numbers:

0.1135, 0.9745, 0.7287, 0.3515, 0.7076, 0.7996

In order to apply the inverse transform method, we need to generate the cumulative distribution function of Y. To do so, we will integrate the PDF:

- $10 \le y < 20$

  $PDF = \dfrac{y-10}{100}$

  $CDF = \displaystyle\int_{10}^{y} \frac{y-10}{100}\, dy = \frac{y^2 - 20y + 100}{200}$

- $20 < y < 30$

  $PDF = \dfrac{30-y}{100}$

$$CDF = \int_{10}^{20} \frac{y-10}{100}\,dy + \int_{20}^{x} \frac{30-y}{100}\,dy = \frac{-y^2+60y-700}{200}$$

Now that we have the CDF, we have to obtain its quantile:

$$Z \sim F(Y)$$
$$Y \sim F^{-1}(Z)$$

- $10 \le y < 20$

$$Z = \frac{y^2-20y+100}{200} \rightarrow y_1 = \frac{20 \pm \sqrt{20^2 - 4*(-200Z+100)}}{2}$$

- $20 \le y < 30$

$$Z = \frac{-y^2+60y-700}{200} \rightarrow y_2 = \frac{-60 \pm \sqrt{60^2 - 4*(200Z+700)}}{2}$$

Bearing in mind that $Z \sim \mathrm{U}[0,1]$, we can obtain the following values for Y.

	$10 \le y < 20$		$20 \le y < 30$	
Z_i	y_{1Ai}	y_{1Bi}	y_{2Ai}	y_{2Bi}
0.1134	14.76	5.24	43.32	16.68
0.9745	23.96	-3.96	27.74	32.26
0.7287	22.07	-2.07	37.37	22.63
0.3515	18.38	1.62	18.61	41.39
0.7076	21.89	-1.89	37.65	22.35
0.7996	22.65	-2.65	23.67	36.33

$$Y \rightarrow [14.76\ 18.38\ 22.35\ 22.63\ 23.67\ 27.74]$$

5. Computer Problem: Consider a distribution with density function as

$$\rho(x) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad 0 \le x \le 1,$$

$$\text{where} B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \Gamma(\cdot) \text{is the gamma function.}$$

Derive an acceptance-rejection method for generating random variables with above pdf with α = 2, β = 2 from U[0, 1] random variables. Generate 1000 such random numbers.

In the acceptance rejection method we want to generate a series of a independent random variables (X1, X2, …), that follow a specific probability density function $\rho(x)$, by using other independent random variables, that follow another PDF $h(x)$.

In order to apply this method, we have to calculate the value of the c constant, which is indeed obtained using the following formula:

$$c^{-1} = sup\ \frac{\rho(x)}{h(x)}$$

In this case, as Z1,Z2.. follow the uniform distribution, we can state that $h(x) = 1$.

Then, we calculate the supremum of $\rho(x)$ by derivating it and finding its maximum in the given range [0, 1].

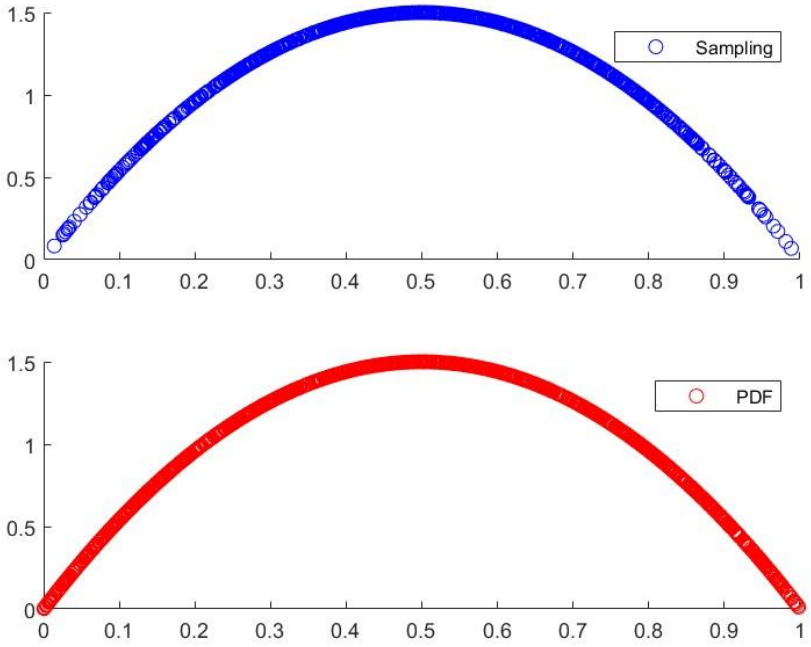
$$\rho(x) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1} = 6 * x * (1-x)$$

$$c^{-1} = sup\frac{\rho(x)}{h(x)} = 6 * 0.5 * (1-0.5) = 3/2$$
$$c\ =\ 2/3$$

Once that we have obtained the efficiency, or c, we can state how many variables we will need in order to achieve the desired number of random variables. In this case, we will need 1.5 times more variables that follow the uniform distribution, than variables with $\rho(x)$ PDF that we want to achieve.
Then, we implement the acceptance-rejection method in Matlab, by using the following code:

```
tic
alfa = 2;
beta = 2;
n = 1e3;

XUV = rand(1.5*n,3); %some uniform random numbers
Y = 1/((gamma(alfa)*gamma(beta)/gamma(alfa+beta)))*((XUV(:,1)).^(alfa-
1)).*(1-XUV(:,1)).^(beta-1);
keep = XUV(:,2) <= (2/3)*Y; %these are the ones that we keep
ZAR = Y(keep); %grab the ones that we want
if length(ZAR)>n
    ZAR = ZAR(1:n); %keep only as many as we need
    XUVb = XUV(keep,1)
    XUVb = XUVb(1:n)
end
toc
figure
tiledlayout(2,1)
ax1 = nexttile;
scatter(ax1, XUVb, ZAR, 'blue')
legend('Sampling');

ax2 = nexttile;
scatter(ax2, XUV(:,1), Y, 'red')
legend('PDF');
```

And the following results are obtained, in which, we can definitely see that the sampled variables follow the PDF with accuracy.


Sampling
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

PDF
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1