University of Southern California

EE511

Simulation Methods for Stochastic Systems

Project #4

Integrals and Intervals

BY

Akshay Deepak Hegde

USC ID: 8099460970

hegdeaks@usc.edu

Question 1:

Approximate the following integrals using a Monte Carlo simulation. Compare your esti mates with the exact values (if known):

a. $\int_{-2}^{2} e^{x+x^2}\,dx.$

b. $\int_{-\infty}^{\infty} e^{-x^2}\,dx.$

c. $\int_0^1 \int_0^1 e^{-(x+y)^2}\,dy\,dx.$

Description:

Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. Monte Carlo methods are mainly used in three distinct problem classes: Optimization, Numerical integration, and generating draws from a probability distribution.

Steps followed:

1. Generate a set of random random numbers u(i) uniformly distributed between 0 and 1.
2. Evaluate the function f(x) at each of these randomly evaluated numbers.
3. The function average is then Monte-Carlo estimate of integration and is given by

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

where N is the number of samples

Random number between 0 and 1 for N samples are generated using rand() function. Then using the substitution method, integrals have been solved such that the limits lie in the range 0 to 1. We can then substitute the value of the obtained random number in this function and the resulting value is stored in 'Result' vector. Now, we take the sum of all the values of Result and divide it by the number of samples. This gives us the Monte Carlo simulation. Also, the theoretical value can be calculated using the inbuilt syms() and int() functions. But this gives us an expression in terms of the exponential. So, we can use double precision to get the numerical value.

CODE:

a)

```
        % Akshay Deepak Hegde  USC ID: 8099460970  %
% ------------------------------------------------------------ %
% Project #4-Integrals and intervals, EE511: Spring 2017
% ------------------------------------------------------------ %
% To approximate integrals using Monte Carlo simulation.
% To compare the estimates with exact values.
% ------------------------------------------------------------ %
clc;
clear;
close all;
% ------------------------------------------------------------ %

n=1000;%Number of samples
for i=1:n
   u(i)=rand();% To generate random numbers
end

for i=1:n %To calculate function inside integral
   Result(i)=4*exp(2-12*u(i)+16*(u(i))^2);
end

E=sum(Result);%To get the sum
monte=E/n;%The Monte Carlo estimation
disp('The Monte Carlo estimate is :')
disp(monte)

%Calculating Theoretically using syms() and int() functions
syms x;
f=exp(x+x^2);
doubleN=double(int(f,x,-2,2));
disp('The theoretical value of the integral is: ')
disp(doubleN)
```

<u>Output</u>

N=100

```
>> ee511_p4q1a
The Monte Carlo estimate is :
  94.8157
```

The theoretical value of the integral is:
  93.1628

N=1000
>> ee511_p4q1a
The Monte Carlo estimate is :
  94.2471

The theoretical value of the integral is:
  93.1628

N=10000
>> ee511_p4q1a
The Monte Carlo estimate is :
  93.5964

The theoretical value of the integral is:
  93.1628

b)

```
n=10000;%Number of samples
for i=1:n
   u(i)=rand();%To generate random numbers
end

for i=1:n %To calculate function inside integral
   Result(i)=(2*exp(-1-1/(u(i))^2+2/u(i)))/(u(i))^2;
end

E=sum(Result);%Sum of the Result
monte=E/n;%The Monte Carlo estimation
disp('The Monte Carlo estimate is :');
disp(monte);

%Calculating Theoretically using syms() and int() functions
syms x;
f=(2*exp(-((1/x)-1)^2))/x^2;
doubleN=double(int(f,x,0,1));
disp('The theoretical value of the integral is: ')
disp(doubleN)
```

<u>Output</u>

N=100
```
>> ee511_p4q1b
The Monte Carlo estimate is :
   1.9549

The theoretical value of the integral is:
   1.7725
```

N=1000
```
>> ee511_p4q1b
The Monte Carlo estimate is :
   1.7562

The theoretical value of the integral is:
   1.7725
```

N=10000
```
>> ee511_p4q1b
The Monte Carlo estimate is :
   1.7713

The theoretical value of the integral is:
   1.7725
```

c)

```
n=100;%Number of samples
for i=1:n
   u(i)=rand();%To generate random numbers
end

for j=1:n %To calculate function inside integral
   I(j)=exp(-4*(u(j))^2);
end

E=sum(Result);%Sum of the Result
monte=E/n;%The Monte Carlo estimation
disp('The Monte Carlo estimate is :');
disp(monte);

%Calculating Theoretically using syms() and int() functions
syms x y;
```

```
f=exp(-(x+y)^2);
doubleN=double(int(int(f,x,0,1),y,0,1));
disp('The theoretical value of the integral is: ')
disp(doubleN);
```

<u>Output</u>

N=100

```
>> ee511_p3q1c
The Monte Carlo estimate is :
   0.4217

The theoretical value of the integral is:
   0.4118
```

N=1000

```
>> ee511_p3q1c
The Monte Carlo estimate is :
   0.4147

The theoretical value of the integral is:
   0.4118
```

N=10000

```
>> ee511_p3q1c
The Monte Carlo estimate is :
   0.4112

The theoretical value of the integral is:
   0.4118
```

Analysis:

It is evident from the above outputs, theoretical values and the Monte Carlo estimates are nearly the same. The theoretical values are calculated by evaluating the integrals. The Monte Carlo estimates are done by averaging over a large number of samples like 100, 1000 and 10000 as shown above. Hence, we can say that Monte Carlo simulation gives us a satisfactory value estimation of the integrals when compared with the theoretical value of the same.

Question 2:

Define the random variable $X = Z_1^2 + Z_2^2 + Z_3^2 + Z_4^2$ where $Z_k \sim N(0,1)$. Then $X \sim \chi^2(4)$. Generate 10 samples from X by first sampling Z_i for $i = 1, 2, 3, 4$ and then computing X. Plot the empirical distribution $F_{10}^*(x)$ for your samples and overlay the theoretical distribution $F(x)$. Estimate a lower bound for $\|F_{10}^*(x) - F(x)\|_\infty$ by computing the maximum difference at each of your samples: $\max_{x_i} |F_{10}^*(x_i) - F(x_i)|$. Then find the 25th, 50th, and 90th percentiles using your empirical distribution and compare the value to the theoretical percentile values for $\chi^2(4)$. Repeat the above using 100 and 1000 samples from X.

Description:

The empirical distribution function Fn*(x) converges to the cumulative distribution function (cdf) F(x) with probability one(According to Glivenko-Cantelli theorem). That is, the estimation gets better and better as we increase the number of samples n. We can generate empirical distribution function which is simply a step function that jumps up by 1/n at each of the n samples.

Chi-Squared random Varaible Distributions have been generated with degree of freedom = 4. Using the cdfplot() and the chi2cdf() functions, the empirical and the theoretical distributions of the chi Square Distributions are overlayed. Initially, 100 elements are taken and it is increased to 1000. For calculating the lower bound of the distribution, I used the ecdf() function, which returns the empirical cumulative distribution function evaluated at the points in given distribution.

The maximum of the difference in the two values of the empirical cumulative distribution function and the theoretical distribution, gives us the lower bound. Prctile() function is used to calculate the percentiles. We then calculate the 25^{th}, 50^{th} and 90^{th} percentile and compare them with the theoretical values.

CODE:

```
N = 100;%Number of samples
X = zeros(N,1);%Initializing with zeros
lower = 1:N;
Z = 1:4;

for j = 1:N %Using for loop to generate N samples of X
  for i = 1:4
    Z(i) = randn();%Generating a normal distribution R.V
    X(j) = X(j) + power(Z(i),2);%calculating X
  end
```

```
end

cdfplot(X);% Plotting empirical distribution function
hold on % To plot with an overlay
grid on
X = sort (X);%Sort X

theoritical = chi2cdf(X,4);%To calculate the theoretical CDF
plot(X,theoritical);
hold off

f10 = ecdf(X);%Gives a vector of values of the empirical cdf evaluated at X.
legend('Empirical cdf','Theoretical cdf');
ylim([0 1.1]);

title('Empirical Distribution');
xlabel('x');
ylabel('f(x)');

for k = 1:N
  lower(k) = abs(f10(k)-theo(k)); %To calculate the lower bound
end
lowerbound = max(lower);

disp('The lower bound is: ')
disp(lowerbound)

%To display output and compare
disp('The 25th percentile');
disp('Empirical Distribution');
disp(prctile(f10,25));
disp('Theoretical Value');
disp(prctile(theoritical,25));

disp('The 50th percentile');
disp('Empirical Distribution');
disp(prctile(f10,50));
disp('Theoretical Value');
disp(prctile(theoritical,50));

disp('The 90th percentile');
disp('Empirical Distribution');
disp(prctile(f10,90));
disp('Theoretical Value');
```

```
disp(prctile(theoritical,90));
```

Output

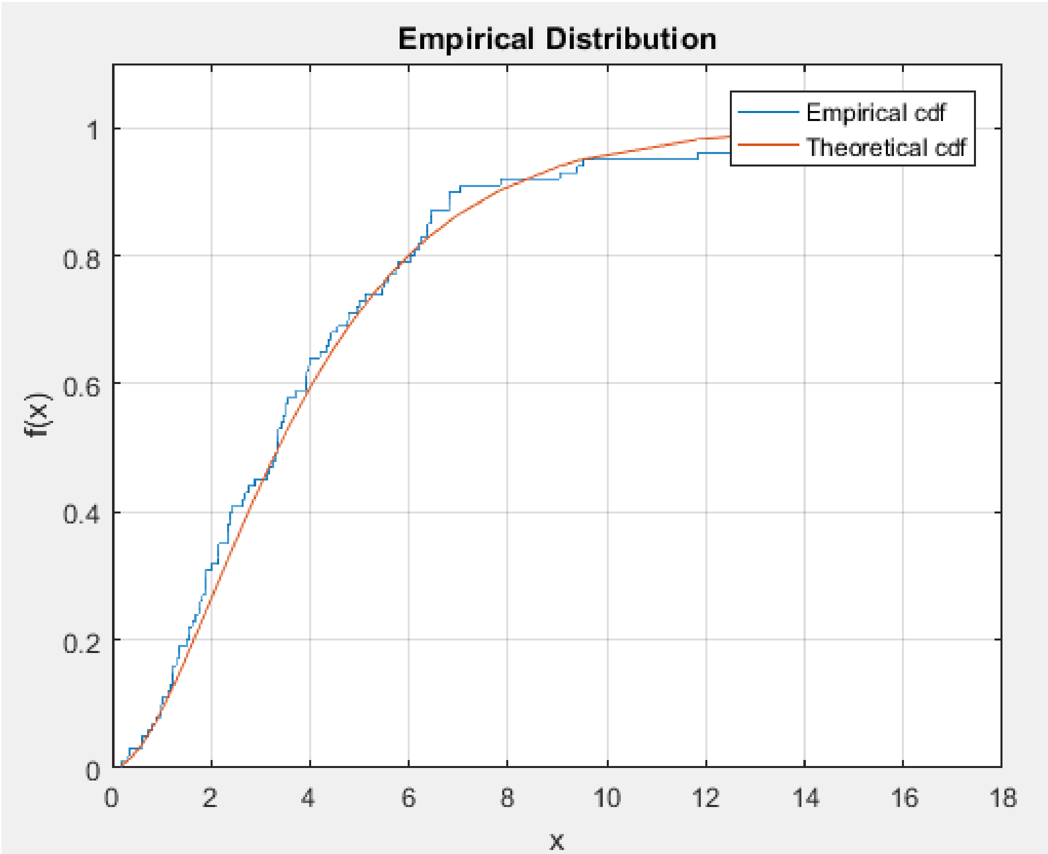
For N=100

```
>> ee511_p4q2
The lower bound is:
    0.0566

The 25th percentile
Empirical Distribution
    0.2475
Theoretical Value
    0.2207

The 50th percentile
Empirical Distribution
    0.5000
Theoretical Value
    0.4962

The 90th percentile
Empirical Distribution
    0.9040
Theoretical Value
    0.8598
```

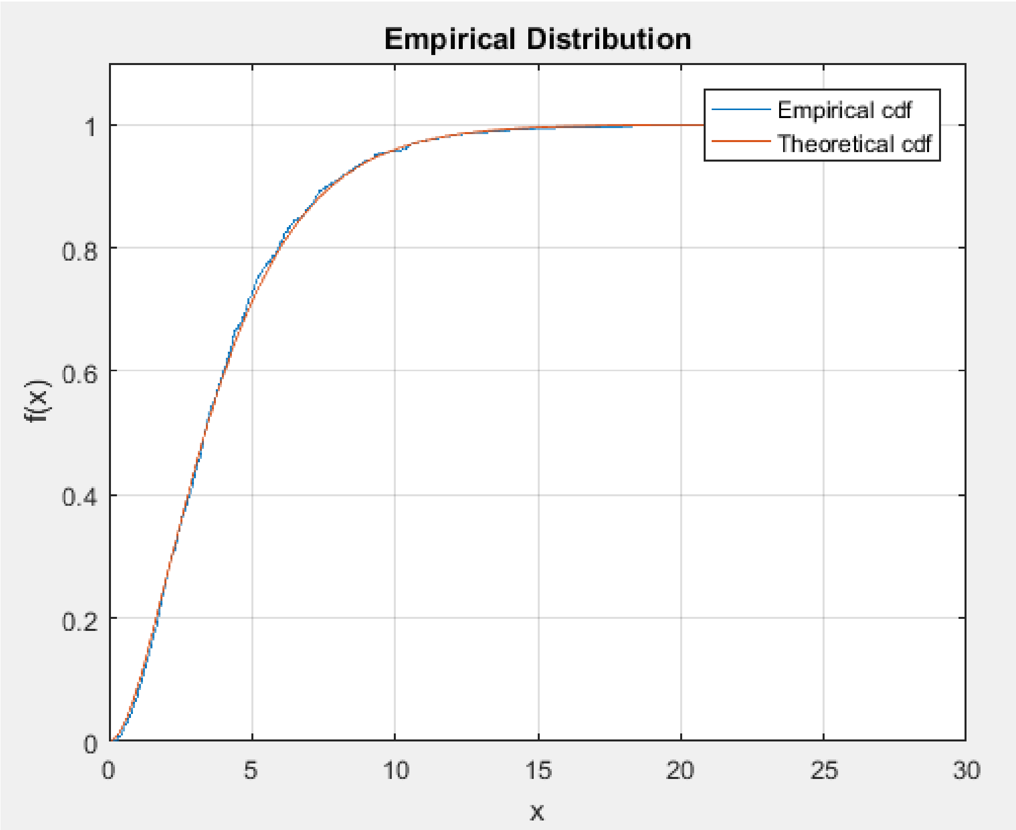


For N=1000
>> ee511_p4q2
The lower bound is:
  0.0190

The 25th percentile
Empirical Distribution
  0.2498
Theoretical Value
  0.2544

The 50th percentile
Empirical Distribution
  0.5000
Theoretical Value
  0.4969

The 90th percentile
Empirical Distribution
    0.9004
Theoretical Value
    0.8935



Analysis:

It is evident from the plots that, the difference between the empirical distribution and the theoretical distribution diminishes considerably when the number of samples is increased. This verifies the Glivenko-Cantelli theorem, according to which the Empirical Distribution Function given by:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{(-\infty,x]}(X_i)$$

converges to a pointwise convergence (F(x)) when $n \rightarrow \infty$
So, the Empirical Distribution Function $F_n(x)$converges to the Cumulative Distribution Function (F(x)), as and when N increases.

Question 3:

A geyser is a hot spring characterized by an intermittent discharge of water and steam. Old Fait hful is a famous cone geyser in Yellowstone National Park, Wyoming. It has a predictable geoth ermal discharge and since 2000 it has erupted every 44 to 125 minutes. Refer to the addendu m data file that contains waiting times and the durations for 272 eruptions. Compute a 95% st atistical confidence interval for the waiting time using data from only the first 15 eruptions. Co mpare this to a 95% bootstrap confidence interval using the same 15 data samples. Repeat t hese calculation using all the data samples. Comment on the relative width of the confidence intervals when using only 15 samples vs using all sample

Description:

A data file is given with 3 columns stating numbers, waiting times and durations of 272 eruptions respectively. The file is read using fileread() function and the values are scanned to "data" variable. Three column values are taken into x, y, and z respectively. Using std() and sqrt(), standard error is calculated and T-score value is calculated using tinv() function. Hence, 95% statistical confidence interval is found out. To calculate the bootstrap confidence interval, bootstrp() function is used. The upper and the lower bounds are found out using prctile() function.

Code:

```
        % Akshay Deepak Hegde  USC ID: 8099460970  %
% ------------------------------------------------------------ %
% Project #4-Integrals and intervals, EE511: Spring 2017
% ------------------------------------------------------------ %
% data file with waiting times and durations are given for 272 eruptions
% To compute 95% statistical confidence interval for first 15 values
% To compute 95% bootstrap confidence interval for first 15 values
% To calculate the same with all data and to compare.
% ------------------------------------------------------------ %
clc;
clear;
close all;
% ------------------------------------------------------------ %
content = fileread( 'faithful.dat.txt' ) ;  % read the data file
data = textscan( content, '%f %f %f%*[^\n]', ...
                 'HeaderLines', 3) ;  % Scan for 3 columns
x = data{1};  % 1st column
y = data{2};  % 2nd column
z = data{3}; % 3rd  column
f = (z(1:15)); % First 15 values
```

```
 % Statistical confidence interval for 15 values
SError = std(z)/sqrt(length(z));            % To calculate standard error
tscore = tinv([0.025  0.975],length(z)-1);     % To calculate T-score
ConfInt = mean(z) + tscore*SError; % Confidence Interval
disp('Statistical confidence interval for 15 values');
disp(ConfInt);

% Statistical confidence interval for all 272 values
SError = std(f)/sqrt(length(f));            % To calculate standard error
tscore = tinv([0.025  0.975],length(f)-1);     % To calculate T-score
ConfInt = mean(f) + tscore*SError; % Confidence Interval
disp('Statistical confidence interval for all 272 values');
disp(ConfInt);

% Bootstrap confidence interval
y = bootstrp(15, @mean, z)  % Bootstrap interval for 15 values
Sort = sort(y)
disp('Bootstrap confidence interval for 15 values');
CIlow=prctile(Sort,2.5)  % Lower bound
CIhigh=prctile(Sort,97.5) % Upper bound

y = bootstrp(272, @std, f) % Bootstrap interval for all 272 values
Sort = sort(y)
disp('Bootstrap confidence interval for all 272 values');
CIlow=prctile(Sort,2.5) % Lower bound
CIhigh=prctile(Sort,97.5) % Upper bound
```

Result:

```
Statistical confidence interval for 15 values
   62.5571   79.3096



Statistical confidence interval for all 272 values
   69.2742   72.5199


Bootstrap confidence interval for 15 values

CIlow =

   62.5333
```

CIhigh =

79.4667

Bootstrap confidence interval for all 272 values

CIlow =

69.1386

CIhigh =

72.4463

Analysis and Discussion:

It is evident from the output that, as the number of samples increases the confidence interval is reduces. When we take 15 samples and compute the statistical and bootstrap confidence intervals, we can

see that the relative width is higher. When we take all samples the relative widths of statistical and bootstrap confidence intervals are almost the same.