EE 511 – Project #2 and #3 Muthulakshmi Chandrasekaran

October 9,2017 USC ID 4486180242

1.1 Problem Statement

To evaluate problems on Monte Carlo Approach.

Part A: To estimate the value of pi using Monte Carlo approach. To find the confidence intervals based on the estimator. To find the number of points needed for the value of pi to be ±1% of the actual value of pi.

Part B: To evaluate the integral $I(n) = \int_{(n-1)\pi}^{n\pi} \frac{\sin(x)}{x} dx$ using Monte Carlo Approach.

Based on the above, to integrate the integral $D(n) = \int_{0}^{n\pi} \frac{\sin(x)}{x} dx$ for different values of n.

To estimate the value of Dirichlet Integral $\int_{0}^{\infty} \frac{\sin(x)}{x} dx$.

Part C: To find out the possibilities of different hands in a Poker Game using Monte Carlo Approach.

1.2 Mathematical Basis

Monte Carlo methods are computational algorithms that use sampling recursively to obtain some numerical results. These algorithms are based on randomness. These algorithms are approximations, but tend to become closer to the actual value when sampling is very large. These methods are used mainly for integration, optimization and probability distributions. Generally, Monte Carlo method are used to solve problems with probabilistic notions. These methods usually follow the steps outlined below.

i. Problem to be defined and the range of possible inputs are chosen
ii. Generate the inputs randomly in the range using a well – defined probability distribution, like uniform or normal
iii. Determine the value using required computations
iv. Repeat for large samples, and aggregate the results.

1.3 Simulation in Matlab

The three problems – Estimation of value of pi, estimation of the value of integrals and finding the probability of hands on a Poker Game are all coded in Matlab.

1.3.1 Part A: Estimation of the value of pi

The value of pi can be estimated by Monte Carlo Approach. The following steps are done to estimate the value of pi.

i. Draw a square of side r. Let the area of the Square be $S\ = r^2$.

ii. Inscribe a circle inside the square. i.e. of radius $r/2$. Then, the area of the circle is C = $\frac{\pi r^2}{4}$.

iii. Scatter points uniformly inside the square.

iv. Count the number of points inside the circle N_i and outside the circle N_O.

v. The ratio $\frac{N_i}{N_o}$ gives an estimate of the ratio of two areas, which approximately equals $\frac{\pi}{4}$ as the number of points increases. Multiply by 4 to get the actual estimate of π .

Description of the Code:

First the number of samples, n is obtained from the user. Then a square of side 1 unit is generated. Then a circle of radius 0.5 units is generated such that the circle is inscribed in the square. Then the n points are scattered randomly in the square. Then the number of points lying inside the circle are found as c. Then $n - c$ gives the number of points outside the circle, but inside the square. The ratio of $\frac{c}{n}$ gives an approximate value of $\frac{\pi}{4}$. Four times c gives an approximate value of π. Figure 1 shows the Command Window Outputs for $n =\ 1, n = 100, n = 1000, n = 10000\ and\ n = 100000$.

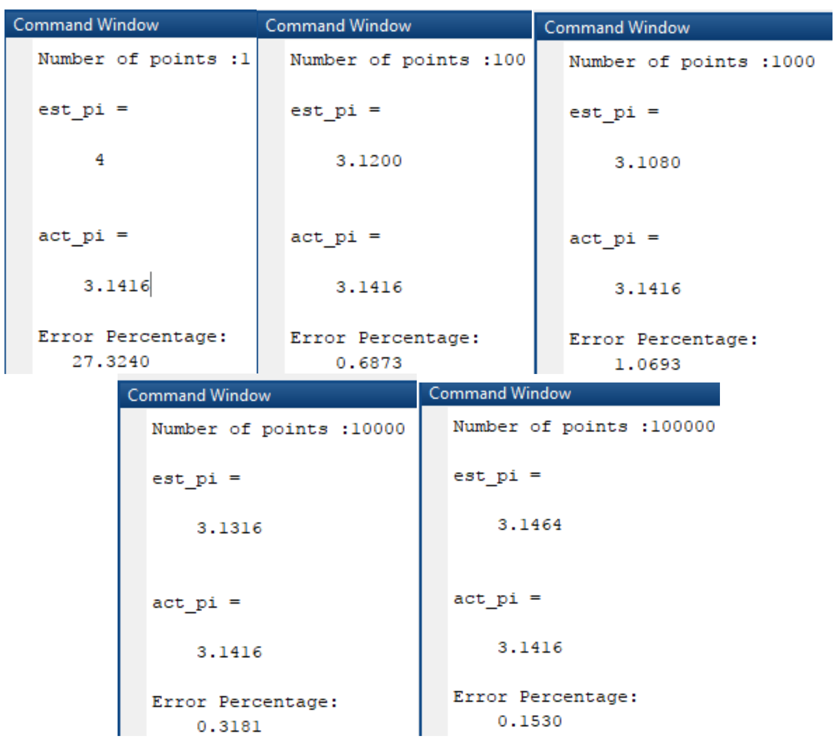

Figure 1 Command Window outputs for estimating the value of pi for different n

Observations:

Figure 2(a) gives a picture of the output of the number of points that is scattered over the square for 1000 sample points. The points become dense as the number of points increases, as shown in Figure 2(b).

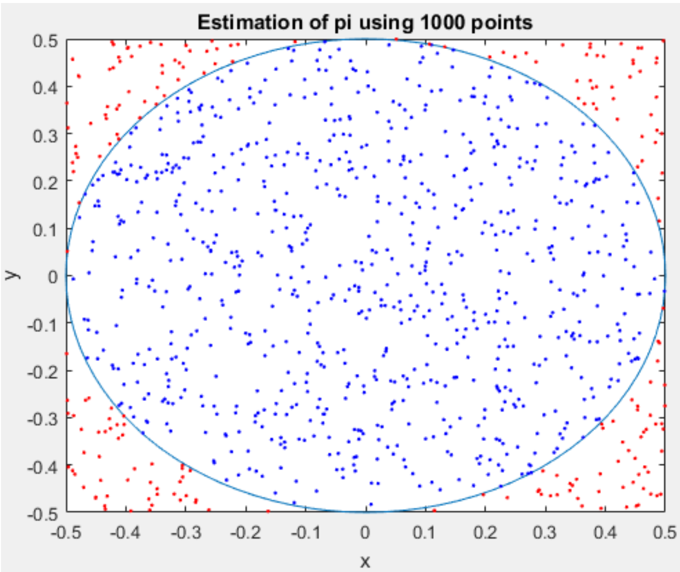

Figure 2(a) Scatter plot for 1000 points

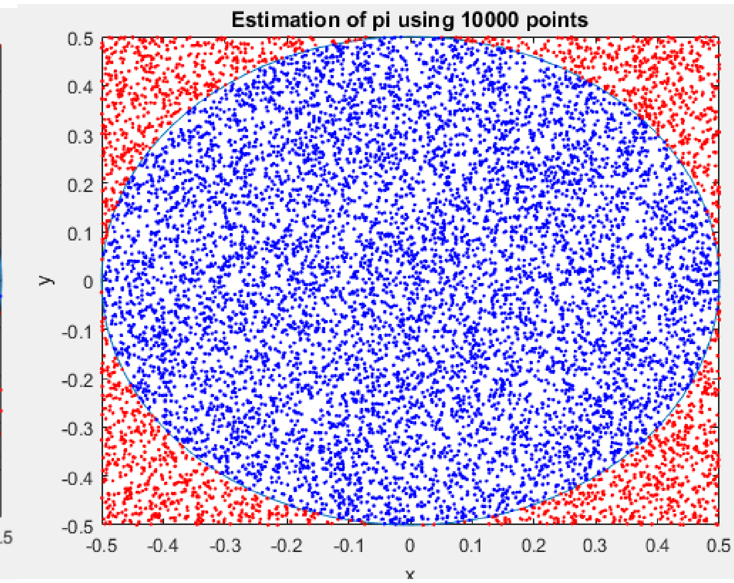

Figure 2(b) Scatter plot for 10000 points

The value of pi when n ranges from 1 to 100000 is plotted as a graph, as in Figure 3.

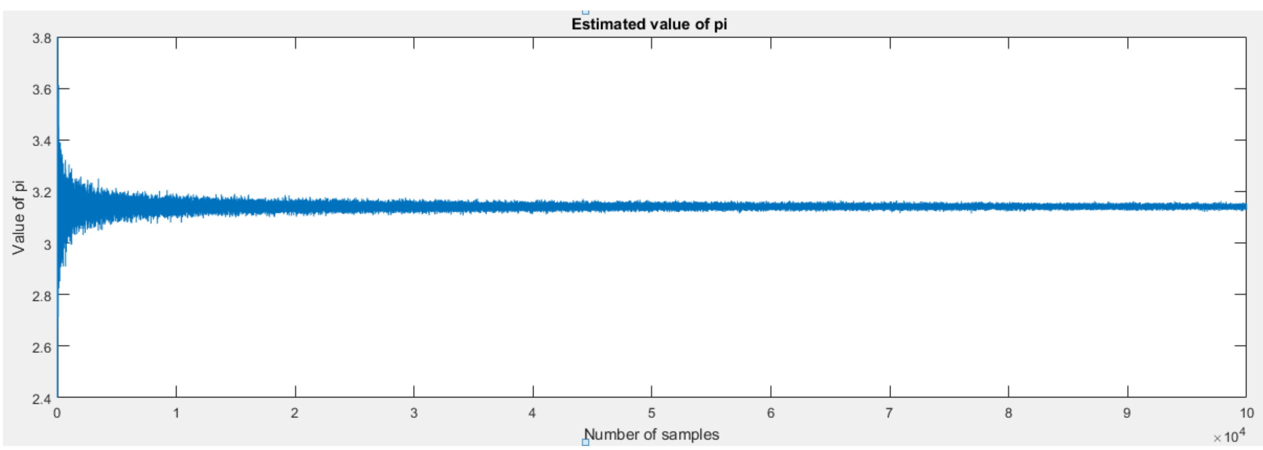

Figure 3 Estimated values of pi for increasing n.

Taking a closer look at the samples, as in Figure 4, it is observed that that as n increases, the estimated value of pi approaches the actual value of pi.

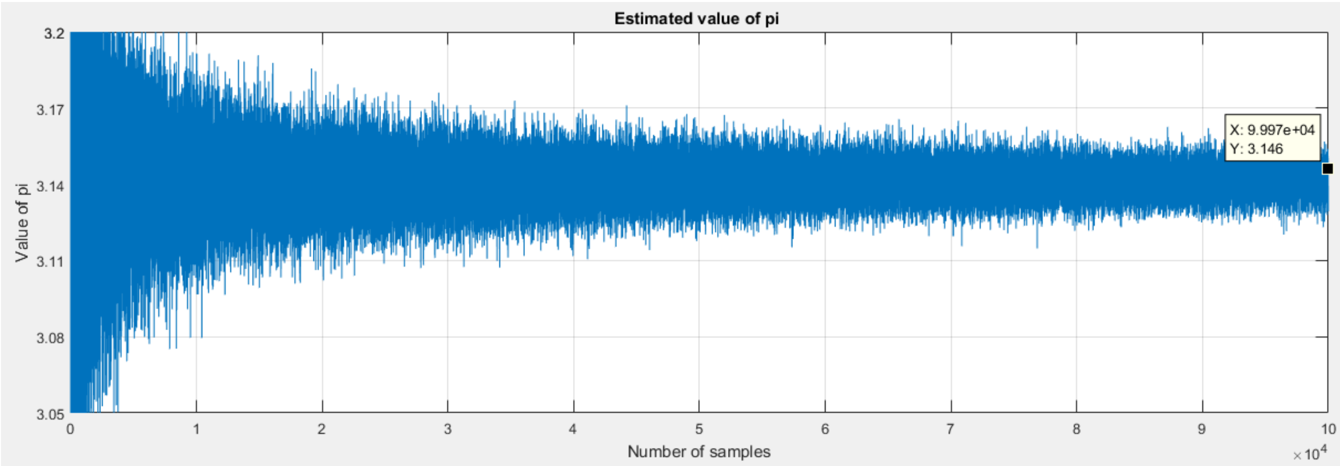

Figure 4 Closer look of the estimated pi values.

Finding Confidence Intervals :

Let $P_i = \begin{cases} 1 & \text{if } (X_i, Y_i) \text{ is in the quadrant} \\ 0 & \text{otherwise} \end{cases}$

P_i follows a Bernoulli Distribution.

$$Prob(P = 1) = p = \frac{\pi}{4} = 0.7854$$

Expected value of P $E(P) = p = 0.7854$

Variance of P $Var(P) = p(1-p) = 0.7854 * (1 - 0.7854) = 0.1685$

Standard Deviation of p $\sigma = \sqrt{Var(P)} = 0.4105$

An estimate $\widehat{p}$ of pi is done. Therefore $\hat{p} = \frac{\sum_{i=1}^{n} P_i}{n}$

It is assumed that $\hat{p}$ follows a Gaussian distribution. Confidence Intervals can be found as

$$Prob\left(p - \beta\sigma_{\hat{p}} \leq \widehat{p} \leq p + \beta\sigma_{\hat{p}}\right) = 1 - \alpha$$

Where β is selected such that probability of $1-\alpha$ meets the desired confidence levels. Given that 0.95 probability is needed. So $1 - \alpha = 0.95$. Therefore $\alpha = 0.05$. The value of $\beta = 1.96$ from the standard Normal Distribution Table. So, therefore

$$Prob\left(3.14 - \frac{1.96 * 0.4105}{\sqrt{n}} \leq \widehat{p} \leq 3.14 + \frac{1.96 * 0.4105}{\sqrt{n}}\right) = 0.95$$

The number of points n is calculated using the formula

$$n = \left(\frac{\beta\sigma}{error\ margin}\right)^2 = \left(\frac{1.96 * 0.4105}{0.01}\right)^2 = 6473$$

Therefore substituting n= 6473,

$$Prob\left(3.14 - \frac{1.96 * 0.4105}{\sqrt{6473}} \le \hat{p} \le 3.14 + \frac{1.96 * 0.4105}{\sqrt{6473}}\right) = 0.95$$

Or

$$Prob(3.1299 \le \hat{p} \le 3.150) = 0.95$$

Therefore the number of points needed for the value of pi to be ±1% of the desired confidence interval i.e (3.129,3.150) is $n = 6473$.

Result:

By Monte Carlo Approach, it is shown that as the number of samples increase, the estimated value of pi approaches the actual value of pi.

1.3.2 Part B – Evaluation of the Integrals

To evaluate the integral $I(n) = \int_{(n-1)\pi}^{n\pi} \frac{\sin(x)}{x} dx$ using Monte Carlo Approach.

Description of the Code:

The entire plot is drawn. The total number of random points,p_{tot} are generated. Then each point is checked to find if it lies under the curve. The total number of points under the curve p_c is calculated. The ratio $\frac{p_c}{p_{tot}}$ is calculated. The product $\frac{p_c}{p_{tot}} * total\ area\ of\ the\ square\ plot$ gives an approximate value of the integral. This is extended for the integrals $D(n) = \int_0^{n\pi} \frac{\sin(x)}{x} dx$ and Dirichlet Integral $\int_0^{\infty} \frac{\sin(x)}{x} dx$.

Observations:

Figure 5 shows the command window screenshots of the integral $I(n) = \int_{(n-1)\pi}^{n\pi} \frac{\sin(x)}{x} dx$ for $n = 1, n = 2, n = 3, n = 4\ and\ n = 5$.

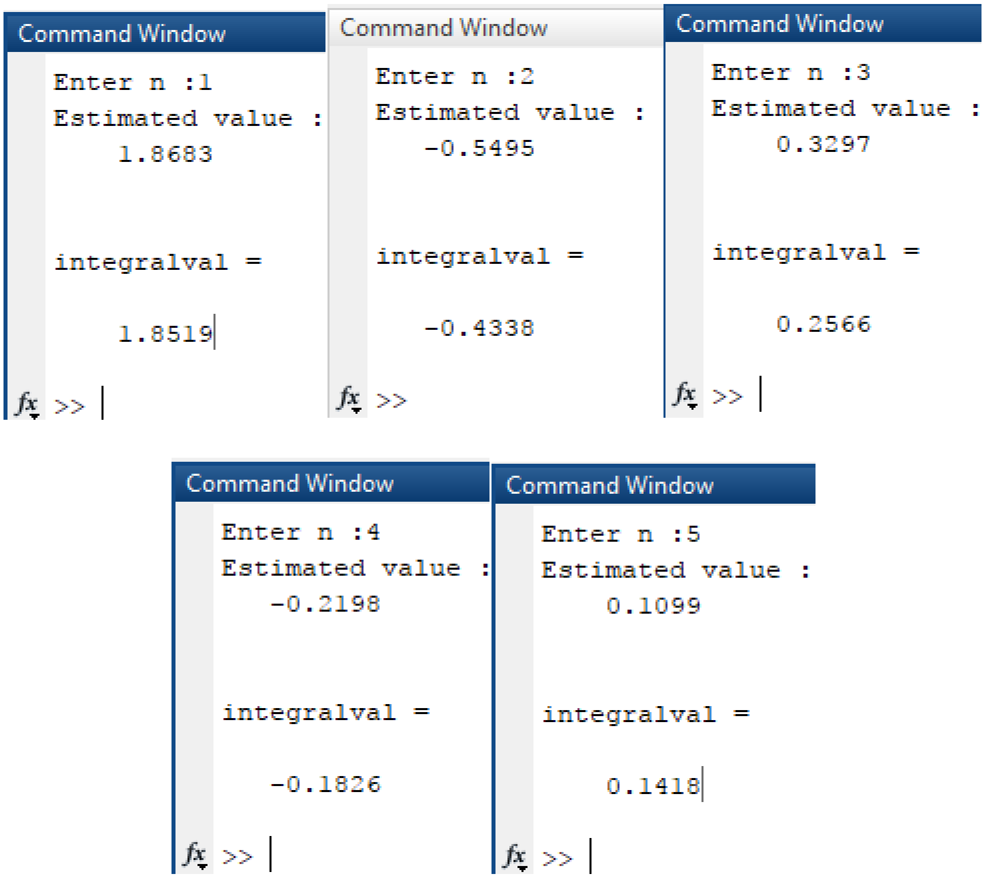

Figure 5 Outputs for the integral $I(n) = \int_{(n-1)\pi}^{n\pi} \frac{\sin(x)}{x} dx$

It is observed that the values of the integrals are closer to each other. Here N=100 points are taken for calculation. If the number of points N is increased, the approximations will be closer to the actual values.

Now, extending the above integral to suit $D(n) = \int_{0}^{n\pi} \frac{\sin(x)}{x} dx$, the lower limit is set to 0 and the upper limit is set to the desired value of $n * pi$. For N=100 points, $n = 10, n = 100 \; and \; n = 1000$ are given, and the outputs are obtained as in Figure 6.

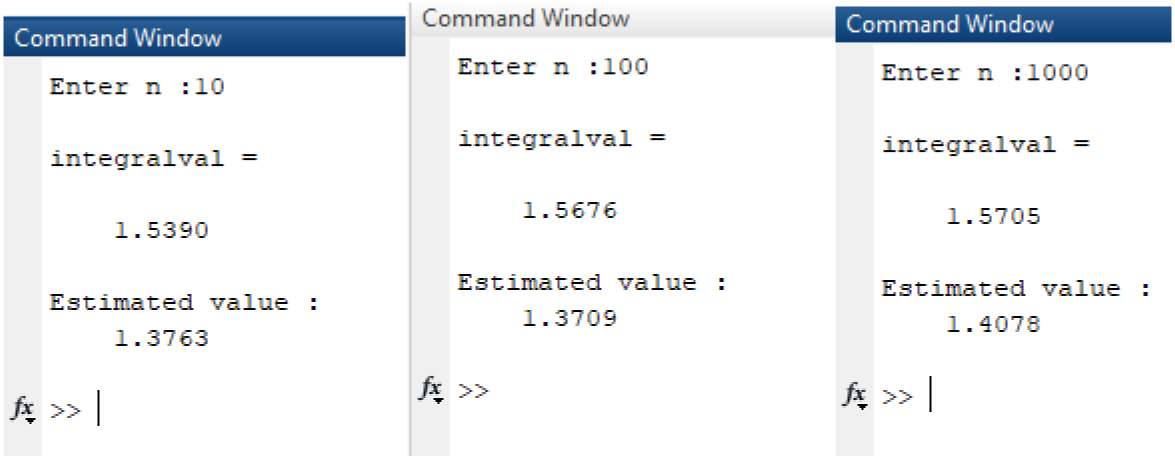

Figure 6 Outputs for the integral $D(n) = \int_0^{n\pi} \frac{\sin(x)}{x} dx$, for n=10, 100 and 1000

It is observed that the values are closer to the actual values, and the error percentage of these values is less than 5%. If the number of random points chosen are increased, the error percentage could be reduced still. Figure 7 shows a sample output for n=10.

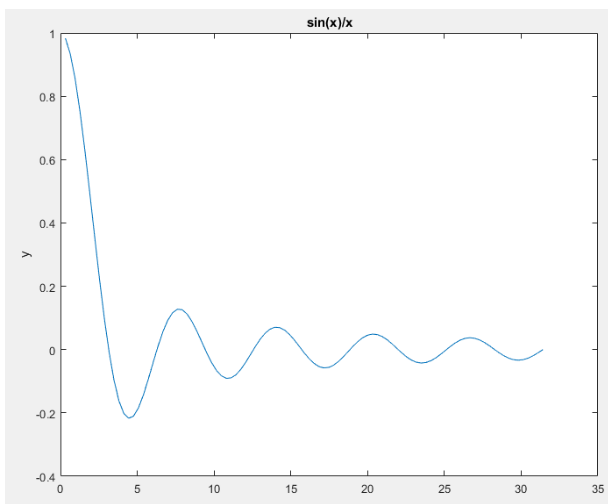

Figure 7 Plot of the curve $D(n) = \int_0^{n\pi} \frac{\sin(x)}{x} dx \;\; for \; n = 10$

Extending this again to the DIrichlet integral $\int_0^{\infty} \frac{\sin(x)}{x} dx$, the upper limit has to be changed to infinity. When this is simulated on Matlab, the output is observed as in Figure 8.

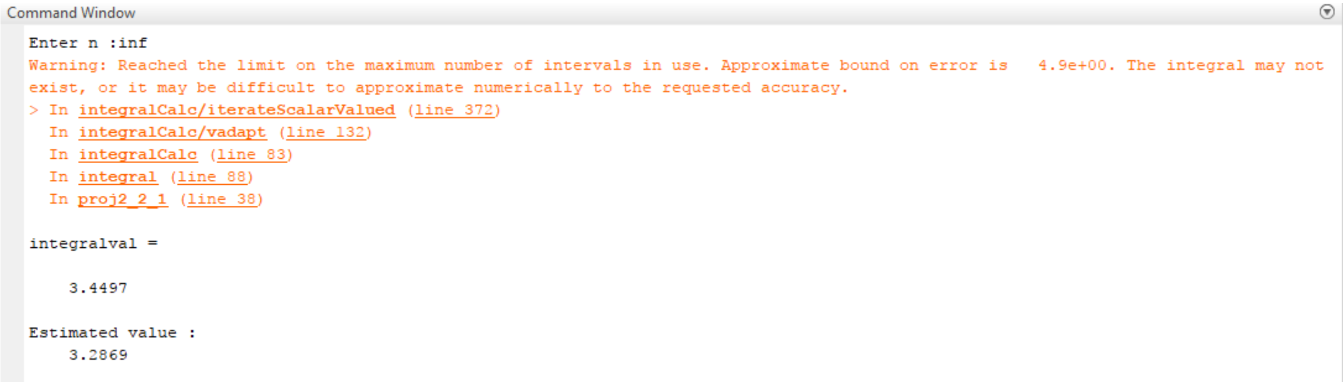

Figure 8 Output for the Dirichlet Integral

Result:

It is observed that for the values approximate the actual values of the integrals, for N=100 random points. If the number of points are increased, the area of the curve tends to the actual value of the integral, thus reducing the error percentage.

1.3.3 Part C – Poker Game

Poker is a game that involves the use of a 52-card deck of playing cards. There are four suits in a deck – Diamonds, Hearts, Clubs and Spades. There are 13 cards (or ranks) in each suit – Ace,2,3,4,5,6,7,8,9,10, Jack, Queen and King. In the game of poker, players attempt to assemble the best five-card hand according to the definitions of each hand that can be made. In a poker game, there are a total of ten hands that can be made – Royal Flush, Straight Flush, Four of a kind, Full House, Flush, Straight, Three of a kind, Two Pair, One Pair and High Card. In this simulation, only five hands are simulated.In five-card stud variation of Poker game, each player is dealt five cards. The six hands that were checked are as follows:

i. Straight Flush: all five cards being of the same suit and all in numerical order.
ii. Four of a kind: the hand must include all the cards of one of the 13 available ranks plus one additional card
iii. Flush: A hand that is a flush must consist of all five cards being of the same suit.
iv. Three of a Kind: This hand must consist of three cards being of the same rank with the other two not improving the hand.
v. Two Pair (or a Pair): Two cards of the same rank, and may or may not be another pair, with an additional card.
vi. High Card: The hand must consist of all five cards being unpaired, non-sequential in rank, and not all the same suit.

Description of the Code:

Each card in the deck is assigned a value between 1 and 52. Then the deck is shuffled. The first five cards in the deck are chosen as a hand. Then the conditions are checked if the hand belongs to a straight flush, four of a kind, flush, three of a kind or two pair. If any of the conditions are not met, the hand is categorized as a High Card. This entire game is run for a large number of simulations, say for N=10000. Then the probability of occurrence of each hand is found by dividing the number of occurrences of the hand by total number of simulations, and compared with the actual probability of occurrence.

Observations:

For smaller number of simulations, the estimated and the actual probability differ by a large margin. As the number of simulations increase, the probability becomes closer to the actual value. Figure 9 shows the actual probability of each hand of the poker game. Figure 10(a) shows the estimated probability of each hand when there are 10 simulations.

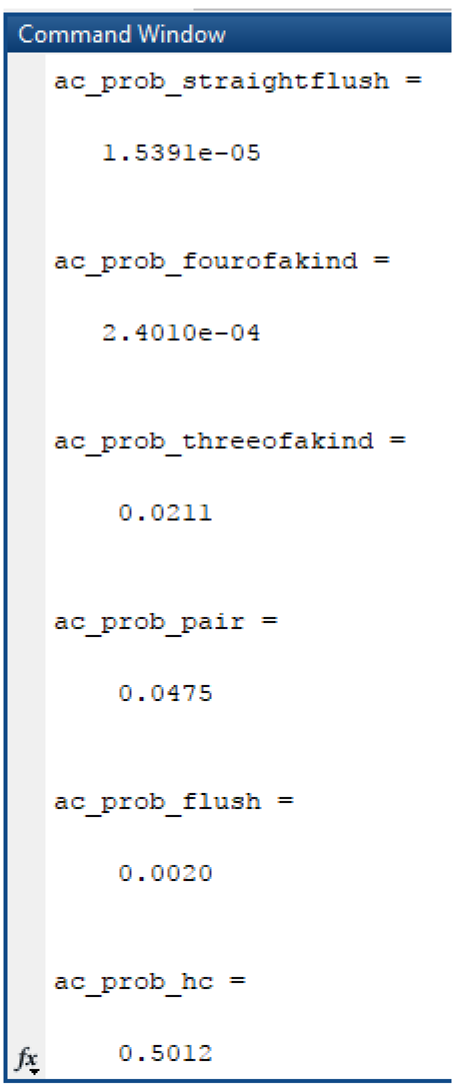

Figure 9 Actual Probabilities of Poker Hands

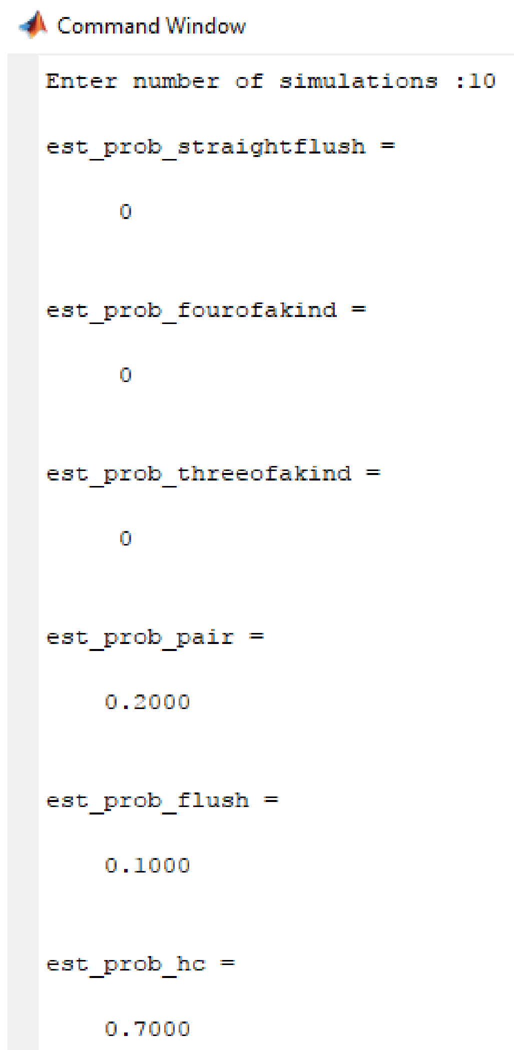

Figure 10(a) Probabilities of Poker Hands when N=10

Figure 10(b), Figure 10(c) and Figure 10(d) show Probabilities of Poker Hands when N=100, N=1000 and N=100000 respectively.

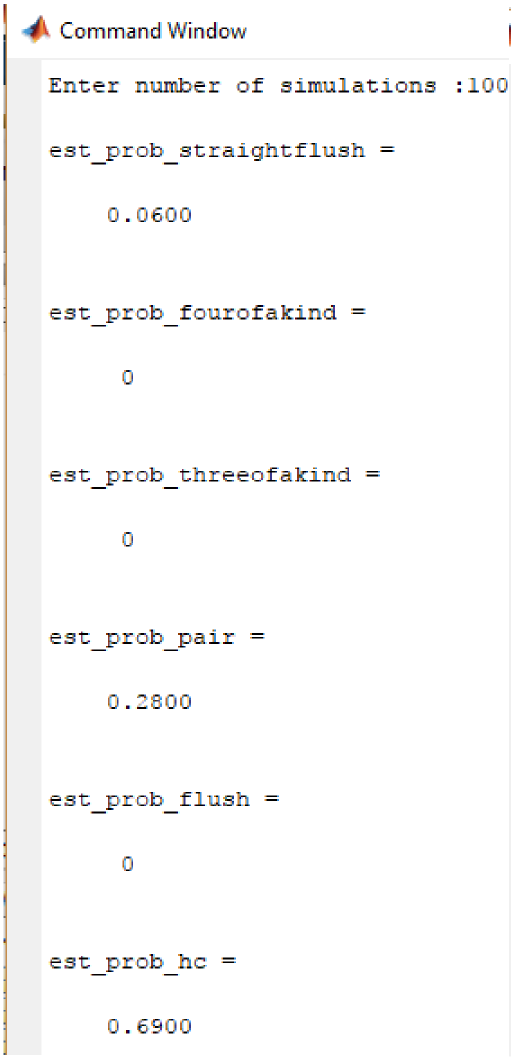

Figure 10(b) Probabilities of Poker Hands when N=100

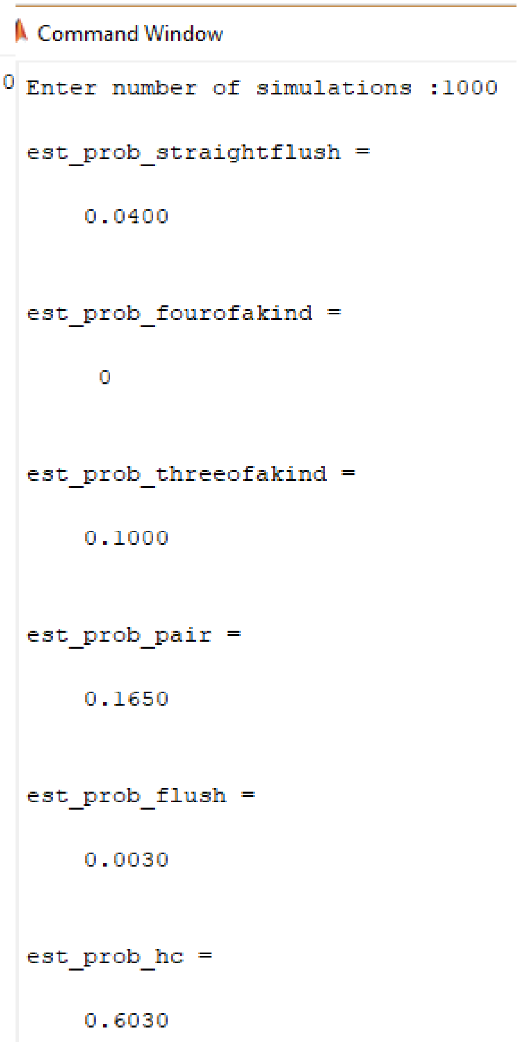

Figure 10(c) Probabilities of Poker Hands when N=1000

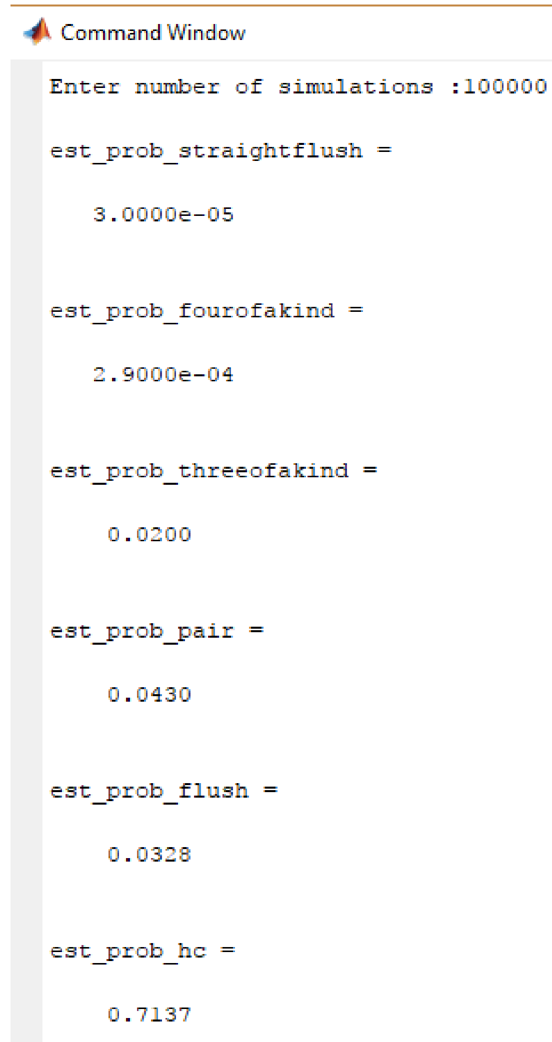

Figure 10(d) Probabilities of Poker Hands when N=100000

The values are found to be approximate as the number of simulations get very large. Moreover, the probability values of the High Card hand don't match because in the Simulation, High Card is chosen as the complement of all the other five hands. In reality, High card is the complement of the other 9 hands.

Result :

Using Monte Carlo approach, the values of probability of six hands of Poker Game are found. It is observed that the estimated probabilities approach the actual values of probability as the number of trials are increased.

1.4 Conclusion

1. Monte Carlo Approach is used for methods involving randomness, and the values are obtained by approximating for a large number of trials. Some basic examples using Monte Carlo approach are simulated.
2. One example is the estimation of the value of pi. When the number of trials become very big, the estimated value of pi is almost equal to the actual value of pi, with a small error percentage.
3. Another example is evaluation of sinc function within limits and evaluation of sinc function on infinite range. Here too, the approximation holds good for large ranges.
4. One more interesting example is the game of Poker. When the simulation is run for large trials, the estimated probability of getting each hand approaches the actual probability.

1.5 Reference

1. Sheldon M Ross, "Simulation", 5th edition
2. www.mathworks.com

1.6 Matlab Codes for the Experiment

Part A : Estimation of pi

```
clc;clear all;close all;
% Get the number of random points
n=input('Number of points :');
% Generate random points using rand
x1=rand(n,1);y1=rand(n,1);

%Generation of a circle
x=x1-0.5;
y=y1-0.5;
t = 0:pi/50:2*pi;
xunit = 0.5 * cos(t);
yunit = 0.5 * sin(t);
%Plot the circle
h = plot(xunit, yunit);hold on

r=(x.^2)+(y.^2);
c=0;
for i=1:n
    %Check if each random point is inside the circle
    if r(i)<=0.25
        %Count the points inside the circle
        c=c+1;
        plot(x(i),y(i),'b.');
    else
        plot(x(i),y(i),'r.');
    end

end
```

```
xlabel('x');ylabel('y'); title(['Estimation of pi using ',num2str(n),' points']);
%Approximate value of pi=( No.of points inside circle/Total no.of points)*4
est_pi=c/(0.25*n);
act_pi=pi
%Calculation of error percentage
diff=(abs(act_pi-est_pi))/act_pi*100;
disp('Error Percentage: ');disp(diff);
```

Part B : Evaluation of integrals

```
clc;clear all;close all;
n=input('Enter n :');
%Define the upper limit of the integral as a and lower limit as b
a=(n-1)*pi;
b=n*pi;
N=100;
t=linspace(a,b,N);

%evaluate the integral sin(x)/x for the range
val=[];
for i=1:length(t)
    val=[val sin(t(i))/t(i)];
end
%plot the curve
plot(t,val)

%Generate two random vectors to choose points
x=(b-a).*rand(length(t),1) + a;x=x';
y=(b-a).*rand(length(t),1) + a;y=y';

c=0;
%Check if the given point is under the curve
for i=1:length(t)
    %Check if x value is within x limits
    if x(i)<=b && x(i)>=a
        for j=1:length(t)
            %Check if y value is within y limits
            if abs(y(i))<=abs(sin(x(i))/x(i)) && abs(y(i))>=0
                plot((sin(x(i))/x(i)),x(i),'b');hold on;
                %Count the number of points
                 c=c+1;
            end
        end
    else
         plot((sin(x(i))/x(i)),x(i),'r');hold on;
    end
end
tot=length(t)*length(t);
%Estimated value = total area of the plot*(no.of points under curve/total
%no.of points)
disp('Estimated value :');
est_val=(3.5*3.14*((tot-c)-tot)/(tot))
fn=@(v) sin(v)./v;
integralval=integral(fn,a,b)
```

Part C : Poker Game

```
clc;clear all;close all;
%Declaring the Deck
```

```
% Deck={Spades(1:13),Clubs(1:13),Diamonds(1:13),Hearts(1:13)}
% Deck ={1,2,3,4,5,6,7,8,9,10,11,12,13     -->Spades
% A,2,3,4,5,6,7,8,9,10,J,Q,K
%     14,15,16,17,18,19,20,21,22,23,24,25,26-->Clubs
% A,2,3,4,5,6,7,8,9,10,J,Q,K
%     27,28,29,30,31,32,33,34,35,36,37,38,39-->Diamonds
% A,2,3,4,5,6,7,8,9,10,J,Q,K
%     40,41,42,43,44,45,46,47,48,49,50,51,52}-->Hearts
% A,2,3,4,5,6,7,8,9,10,J,Q,K
deck=1:1:52;
d1=deck;
%Number of simulations
num_sim=input('Enter number of simulations :');

%declare the count of each hand occurrence
n_sf=0;
n_fk=0;
n_tk=0;
n_pair=0;
n_fl=0;
n_hc=0;
for c=1:num_sim
    s1=0;
    % Shuffle the deck
    for i=1:1:52
        x=randi(52);
        t=d1(i);
        d1(i)=deck(x);
        d1(x)=t;
    end

    % Draw the first five cards
    dr=d1(1:5);

    % Checking for straight flush
sf=1;
for i=1:1:4
    if abs(dr(i)-dr(i+1))==1
        sf=1;

    else
        sf=0;
    end
end
if sf==1
%     disp('Straight flush!!!');
    s1=1;
    n_sf=n_sf+1;
else
%     disp('Not a straight Flush!');
end

% Checking for four of a kind
c=[dr(1)-dr(2) dr(1)-dr(3) dr(1)-dr(4) dr(1)-dr(5) dr(2)-dr(3) dr(2)-dr(4) dr(2)-
dr(5) dr(3)-dr(4) dr(3)-dr(5) dr(4)-dr(5)];
c=abs(c);
y1= (c==13);
if sum(y1(:)==4)
```

```
    n_fk=n_fk+1;s1=1;
%     disp('Four of a kind!');
else
%     disp('Not a four of a kind');
end

%Checking for three of a kind
y2=(c==13);
if sum(y1(:)==2)
    n_tk=n_tk+1;s1=1;
%     disp('Three of a kind!');
else
%     disp('Not a three of a kind');
end

%Checking for Pair
y3=(c==13);
if sum(y3(:)==1)
    n_pair=n_pair+1;s1=1;
%     disp('A Pair!');
else
%     disp('Not a Pair');
end

%Checking for flush
c2=[fix(dr(1)/13) fix(dr(2)/13) fix(dr(3)/13) fix(dr(4)/13) fix(dr(5)/13)];
if all(c2==c2(1))
    n_fl=n_fl+1;s1=1;
%     disp('Flush');
else
%     disp('Not a flush');
end

%High Card
if s1==0
    n_hc=n_hc+1;
end
end
%Estimated probability = no.of occurrences/total no.of simulations
est_prob_straightflush=n_sf/num_sim
est_prob_fourofakind=n_fk/num_sim
est_prob_threeofakind=n_tk/num_sim
est_prob_pair=(n_pair/num_sim
est_prob_flush=n_fl/num_sim
est_prob_hc=n_hc/num_sim
%Actual probability = No.of possible occurrences/total no.of combinations
disp('Actual Probability :')
ac_prob_straightflush=40/2598960
ac_prob_fourofakind=624/2598960
ac_prob_threeofakind=54912/2598960
ac_prob_pair=123552/2598960
ac_prob_flush=5108/2598960
ac_prob_hc=1302540/2598960
```