Project 2: Monte Carlo Methods

EE 511 – Section: Tuesday 5 pm

Name: Darshan Patil

Student ID: 9575227834

1.

i. Problem Statement

Estimate π by the area method including confidence intervals on your estimate. Draw a graph of the successive values of the estimator as the number of samples increases.
How many points do you need to use for your estimate to be within $\pm 1\%$ of the true value of π (with probability 0.95)?

ii. Theoretical Exploration

To find the value of Pi,
Consider a quadrant with center at the origin and radius r=1 and a rectangle enclosing the quadrant. Generate a pair of $U(0,1)\ RV's\ (X_i, Y_i)$. These correspond to points in the square Some will be inside the quadrant $(if\ X_i^2 + Y_i^2 \leq 1)$, some outside. Use the ratios of the areas of the two geometric figures to estimate π.

$$Estimated\ Pi = 4 * \left(\frac{PointsInCircle}{PointsInRectangle}\right)$$

To find the estimated Pi value to be within $\pm 1\%$ of Expected Pi value,

$$p = \frac{\{Estimated\ Pi\ value - Expected\ Pi\ value\}}{Expected\ value}$$

Check if p lies between $\pm 1\%$. We Increase the number of points i.e. $RV's(X_i, Y_i)$ for N=100, N=1,000, N=10,000 and calculate the N for which probability that the Estimated Pi value is within $\pm 1\%$ of Expected Pi value should be 0.95.

Estimate of $P_{est} = \sum_{i=1}^{n} P_i$ and variance is $\sigma_p^2 = \frac{p(1-p)}{n}$
Here we assume the distribution is Gaussian distributed and which is valid asymptotically with the Central Limit Theorem. Therefore, the Confidence Interval is

$$\Pr\{p - \beta\sigma_p \leq p_{est} \leq p + \beta\sigma_{p_{est}}\} = 1 - \alpha$$

iii. Simulation Methodology

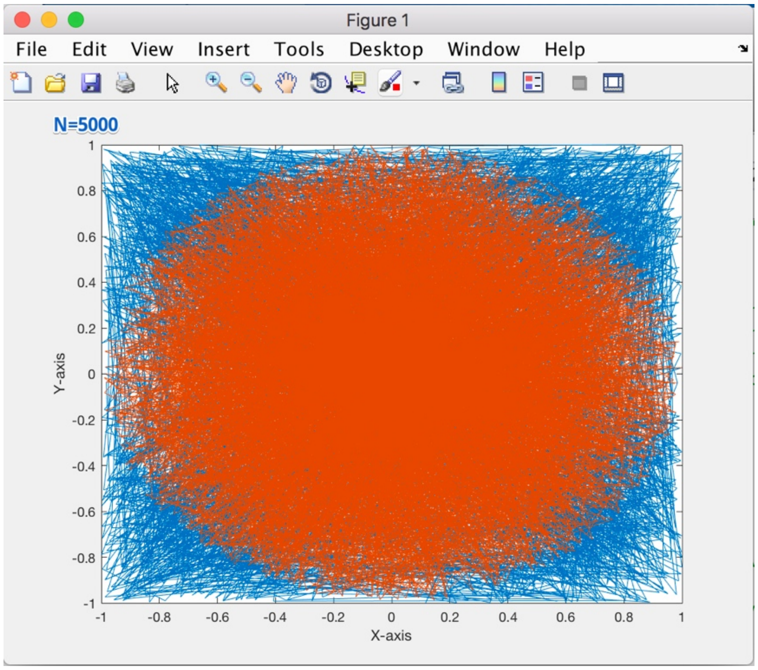
Simulation steps:

- Assign Number of Trials Value.
- Generate N random variable between in the interval (-1,1) for X and Y.
- Check if the points on x-axis and y-axis are within the circle using $(X^2 + Y^2 < 1)$
- Store the points value which are within circle in (X_c, Y_c) matrix.
- Count the number of points in the circle.

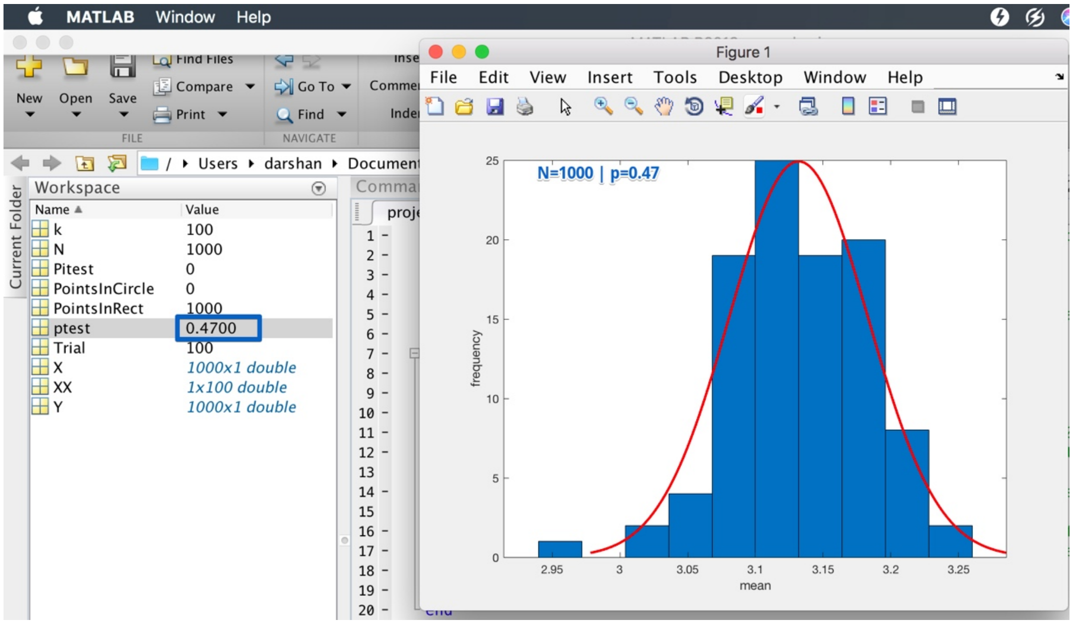
- Calculate the Estimated Pi using the formula mentioned above.
- Check if Estimated Pi lies between $\pm 1\%$ of Expected Pi value.
- Increase the value of N till the probability of Estimated Pi value lies between $\pm 1\%$ of Expected Pi value with 0.95.
- Display the Graphs and values calculated.

iv. Results

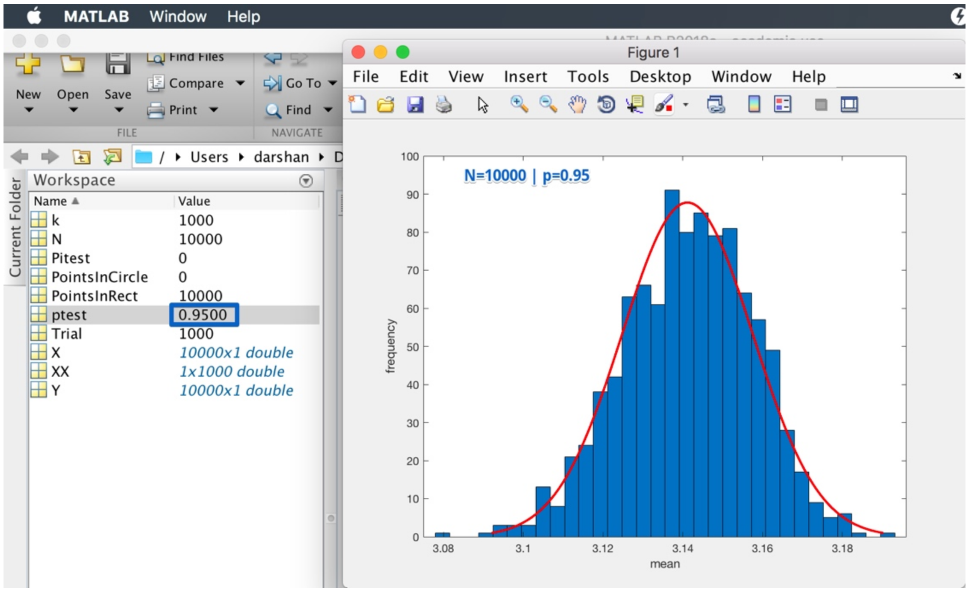
For 5000 Points on rectangle, Estimated Pi value= 3.1496



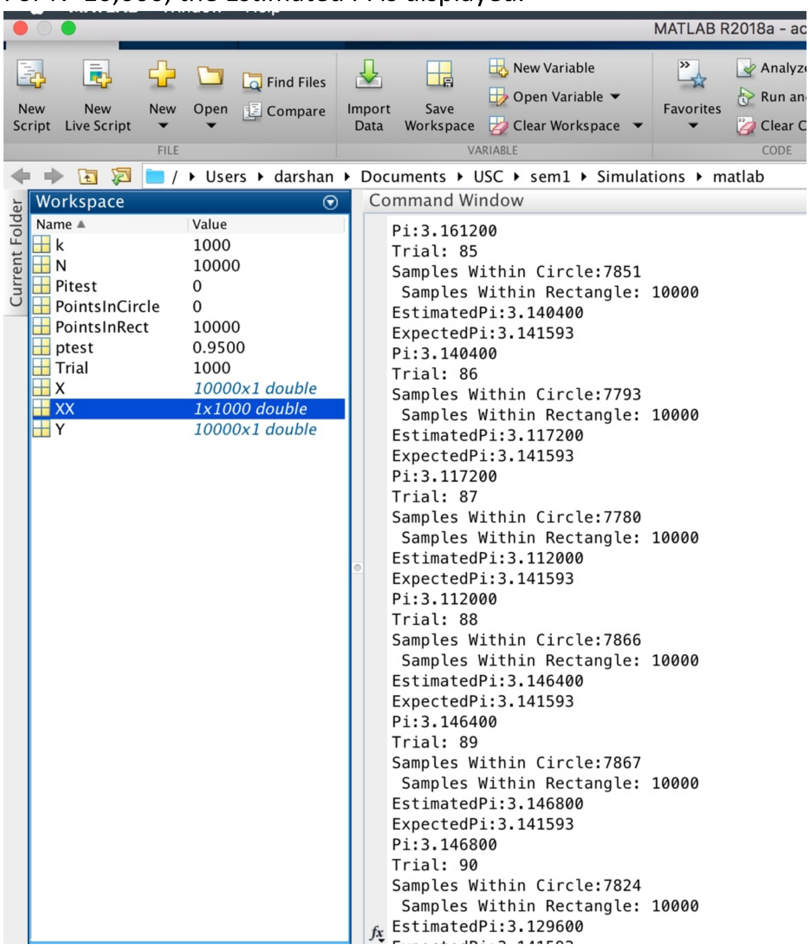
For N=1000, p=0.47



For n=10,000, we get p=0.95, Estimated Value is Normal Distributed for 1000 Trials



For N=10,000, the Estimated Pi is displayed.



Discussion of results:
We have successfully calculated the Estimated Pi value for N=100, 1000, 1000 for different Trials. We observe that Estimated Pi value is between $\pm 1\%$ of Expected Pi value for N=10,000.

v. References

- https://www.tutorialspoint.com/matlab/
- https://www.mathworks.com/help/matlab/ref/integral.html#btdd9x5
- Lecture 5 - Simple Monte-Carlo Methods (2up).pdf by professor Silvester

vi. Source Code

```
clear all;                      % Clear and close matlab, desktop, workspace
close all;
clc;
Trial=100;                        % Number of Trials
XX=zeros(1,Trial);               % XX to store the estimated pi values for each trial
ptest=0;
for k=1:Trial
   N=10000;                     % Number of random number's for each Trial
   X= 1-2*rand(N,1);           % random points on x-axis
   Y= 1-2*rand(N,1);           % random points on y-axis
   PointsInCircle=0;          % Intialize points in circle to 0
   PointsInRect=N;            % Intialize points in rectangle to be N
   %EstimatedPi=0;
   Pitest=0;                          % Intialize Pitest | Pitest is used to check  if estimated pi is within
                                        % +-1% of Expected Pi value
  [XX(1,k),ptest]=et(N,X,Y,PointsInCircle,PointsInRect,k,ptest);
% function Call et to calculate estimated pi for each trial

   if(k==Trial)                  % At the end of Trial compute the percent of
     ptest= ptest/Trial;        % success between confident interval +-1%
   end
end

histfit(XX);                    % Plot Estimated Pi value for various values
xlabel('mean');
ylabel('frequency');

function [EstimatedPi,pp]= et(N,X,Y,PointsInCircle,PointsInRect,k,pp)
  ExpectedPi= pi;              %Assign expected pi value from pi matlab inbuilt function
  Xc=zeros(N,1);
  Yc=zeros(N,1);
for i=1:N
```

```
        if X(i,1)^2+Y(i,1)^2 < 1             % Check if points are within circle
          Xc(i,1)=X(i,1);                    % store x-axis points within circle
          Yc(i,1)=Y(i,1);                    % store y-axis points within circle
          PointsInCircle=PointsInCircle+1;       % count points in circle
        else
          Xc(i,1)=0;
          Yc(i,1)=0;
        end

      if(i==N)
      EstimatedPi=4*(PointsInCircle/PointsInRect);       % Calculate Pi value
      prob=(ExpectedPi-EstimatedPi)/ExpectedPi;         % Check if estimated pi value is between 1% of
                                                          expected pi value
%       if(-0.01 <=prob && prob<=0) || (0<=prob && prob<=0.01)
        if (-0.01<=prob && prob<=0.01)        %check if prob is between +-1% of real pi value
          Pitest=EstimatedPi;              % assign pitest
          pp=pp+1;
        else
          Pitest=0;
        end
%        figure(1);
%        plot(X,Y);                   % Plot points within rectangle
%        hold on;
%        plot(Xc,Yc);                   % Plot points within circle
      fprintf("Trial: %d \nSamples Within Circle:%d\n Samples Within Rectangle:
%d\nEstimatedPi:%f\nExpectedPi:%f\nPi:%f\n", ...
k, PointsInCircle,PointsInRect,EstimatedPi,ExpectedPi,Pitest);
      end
end
end
```

2.

i. Problem Statement

Consider a deck of cards (for simplicity numbered 1…N). Use a uniform random number generator to pick a card and record what card it is (if you were using actual cards, you would replace the card back into the deck – that is not necessary here since we never really take the card out of the deck). Repeat this N times, recording the number of times that each of the cards is selected. Some cards may not show up (actually, it is very likely that several card numbers will not show up and some will show up more than once. You can use this data to estimate the following probabilities:

$P_j = \Pr\{a\ card\ will\ be\ selected\ j\ times\ in\ the\ N\ selections\}$

It is unlikely that any card will show up more than about 10 times. Run this for $N = 10, N = 52, N = 100, N = 1{,}000, N = 10{,}000\ and\ verify\ that\ p_0 \cong \frac{1}{e}$. Can you also find values for the other P_j based on a mathematical analysis?

ii. Theoretical Exploration

Consider a deck of N cards,

Using uniform random number generator, we pick a card and record the value of card.

Generate the N random number between 1 to N, recording the number of times that each of the card is selected.

$P_j = \Pr\{a\ card\ will\ be\ selected\ j\ times\ in\ the\ N\ selections\}$

For N=10, N=52, N=100, N=1,000, and N=10,000 we calculate the p_0 and observe $p_0 \cong \frac{1}{e}$

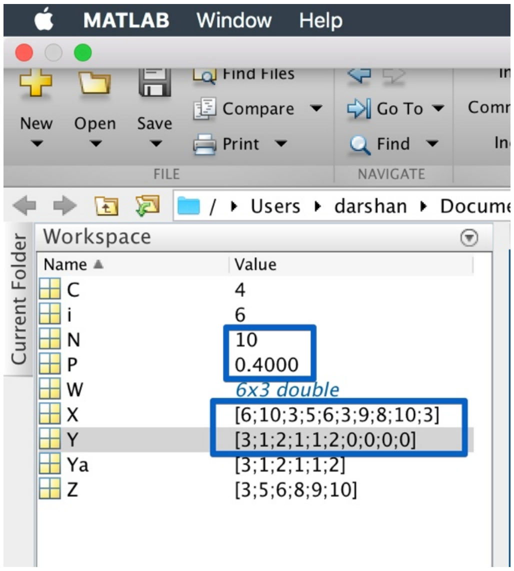
iii. Simulation Methodology

Simulation steps:

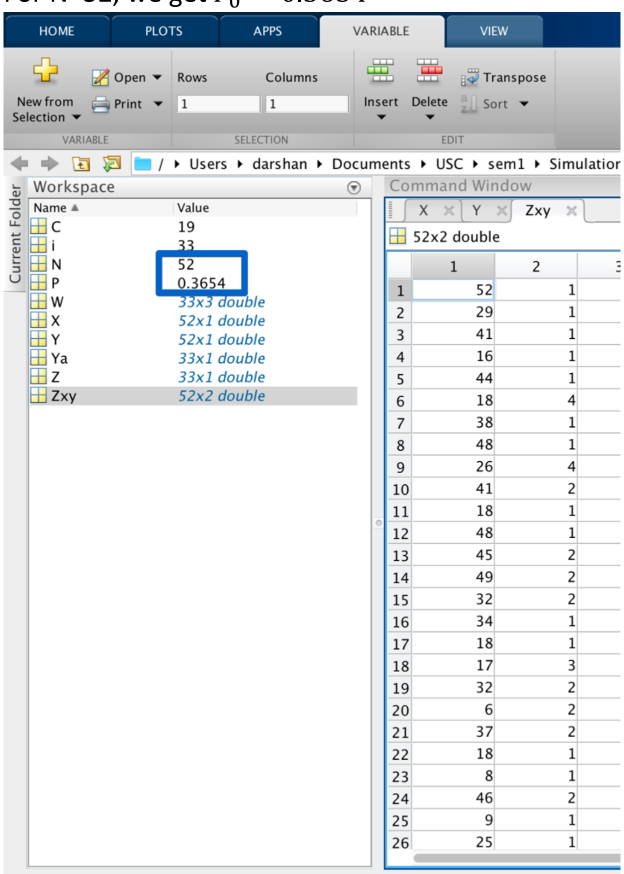
- Assign the Value of N
- Generate N random number between 1 to N and store in X matrix
- Calculate the repeated random number in the Trial using the matlab sum function
- Count the random number's which are not generated in N trial
- Calculate P_0 for N=10, N=52, N=100, N=1000 and N=10,000
- P_0 gets close to $\frac{1}{e}$ as N increases.

iv. Results

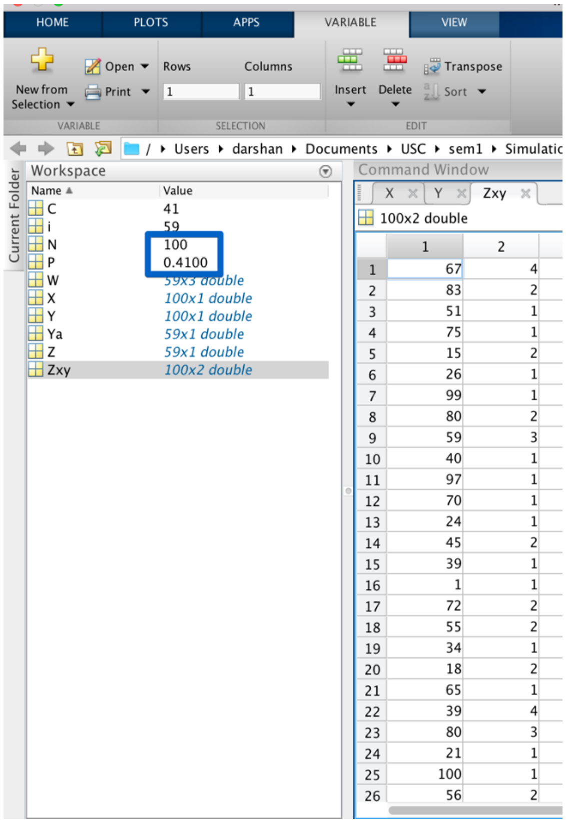
For N=10, we get $P_0 = 0.4$



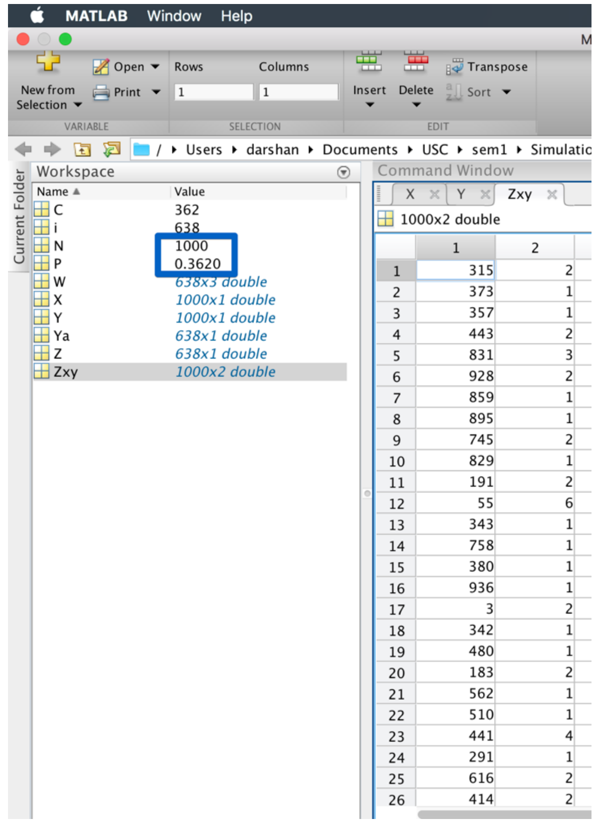
For N=52, we get $P_0 = 0.3654$



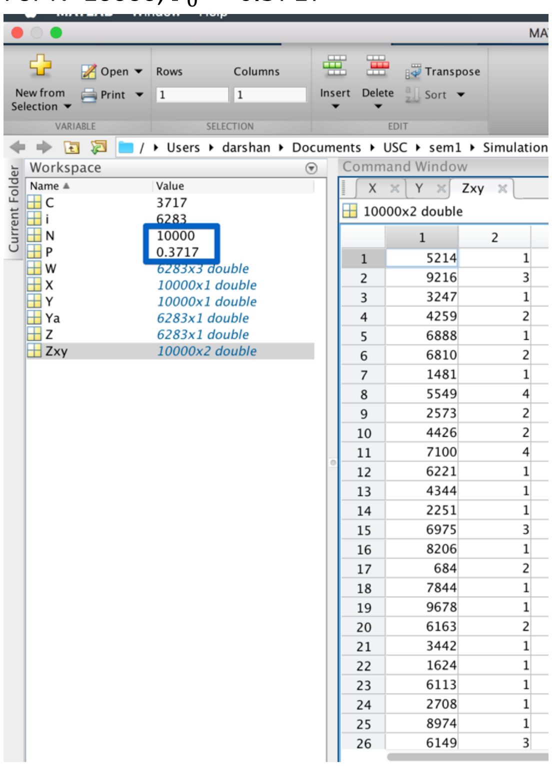
For N=100, $P_0 = 0.41$



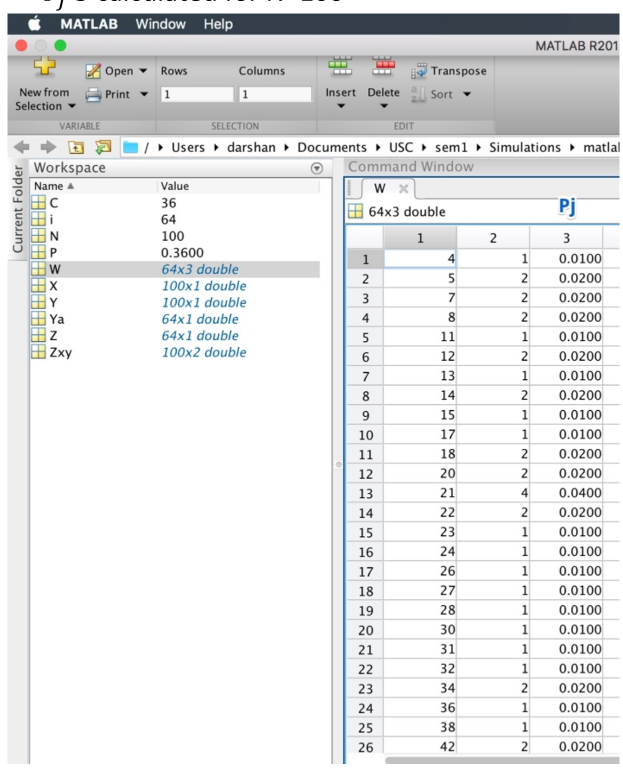
For N=1000, P_0 =0.3620



For N=10000, $P_0 = 0.3717$



P_j's calculated for N=100



Discussion of Results

We calculate the value of P_0 for various values of N, and we observe that as we increase the value of N, P_0 tends to be equal to $1/e$.

v. References

- https://www.mathworks.com/help/matlab/

vi. Source Code

```
clear all; % clear and close matlab, desktop, workspace
clc;
rng = default;            % control the random number generation
N=10;                     % Assign Value of N
X=randi([1,N],N,1);       % Generate N random variable's between 1 to N
Y=zeros(size(X));
Ya=Y;
Z=(unique(X));            %  calculate the unique values in X
W=zeros(length(Z),3);
for i=1:length(Z)
    Y(i,1)=sum(X==Z(i,1));    % calculate the repeated Random numbers
end
C=sum(Y(:)==0);           % count the number of zero's for number not
                                generated in RV
Ya=Y(1:length(Z),1);
W=[Z,Ya,Ya/N];
P=C/N;                    % Calculate P0 which approximates to 1/e
```

3.

i. Problem Statement

Use the method discussed in class to find $\hat{y}$, an estimate for Y and find a 95% confidence interval for the value of the integral.

$$Y = \int_0^{\pi} \frac{Sin(x)}{x} dx$$

ii. Theoretical Exploration

$$I = \int_0^{\pi} \frac{sin(x)}{x} dx$$

Substitute $y = \frac{x}{\pi},\ \ i.e. x = \pi y\ \ and\ dx = \pi dy, so\ integral\ becomes$

$$I = \int_0^{1} \frac{\sin(\pi y)}{\pi y} \pi dy = \int_0^{1} \frac{\sin(\pi y)}{y} dy$$

$$I = \int_0^{n\pi} \frac{sin(x)}{x} dx\ \rightarrow 0\ \ as\ \ \ n \rightarrow \infty$$

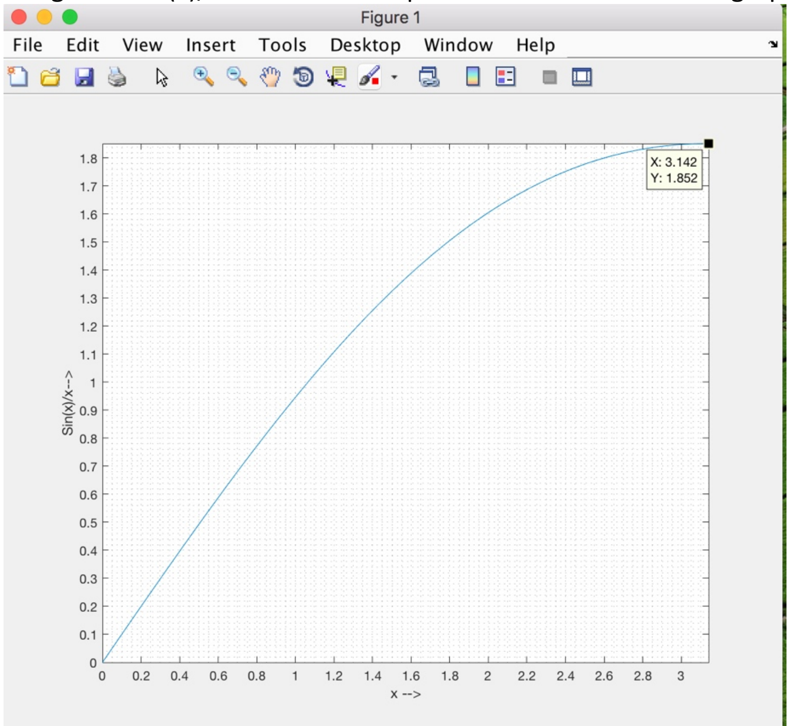
- For a 95% confidence interval, the area in each tail is equal to 0.05/2=0.025. If the area is 0.025 , the value z* such that P(Z>z*)=0.025 or P(Z<z*)=0.975, is equal to 1.96 .

iii. Simulation Methodology

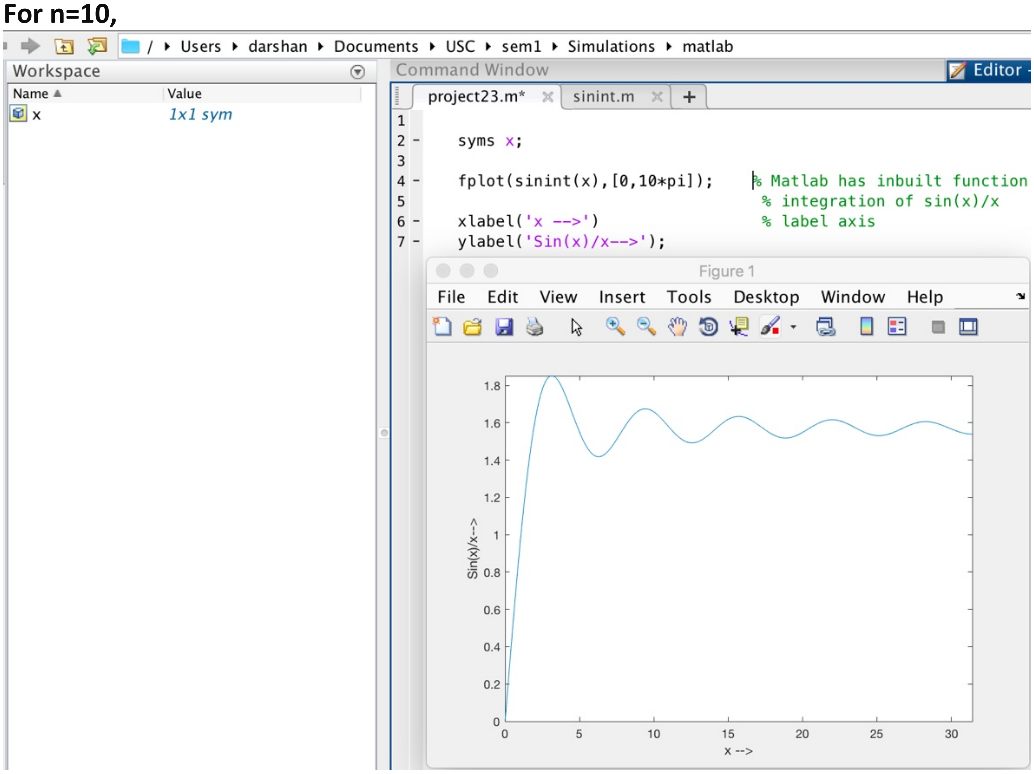
- Use the MATLAB inbuilt function to integrate sin(x)/x from 0 to pi we analyse the integral for different values of n.
- Compute the value for $\frac{\sin(\pi x)}{x}$ by iterating between j=1 to j=Trials.
- Perform summation of $\frac{\sin(\pi x)}{x}$ values which were generated.
- The theoretical value can be obtained using the matlab in-built in function.
- The confidence interval for 95% value is computed using the formula
  $i1 = X1 - \frac{1.96 * X2}{sqrt(trial)}\ and\ i2 = X1 + \frac{1.96 * X2}{sqrt(trial)}$

iv. Results

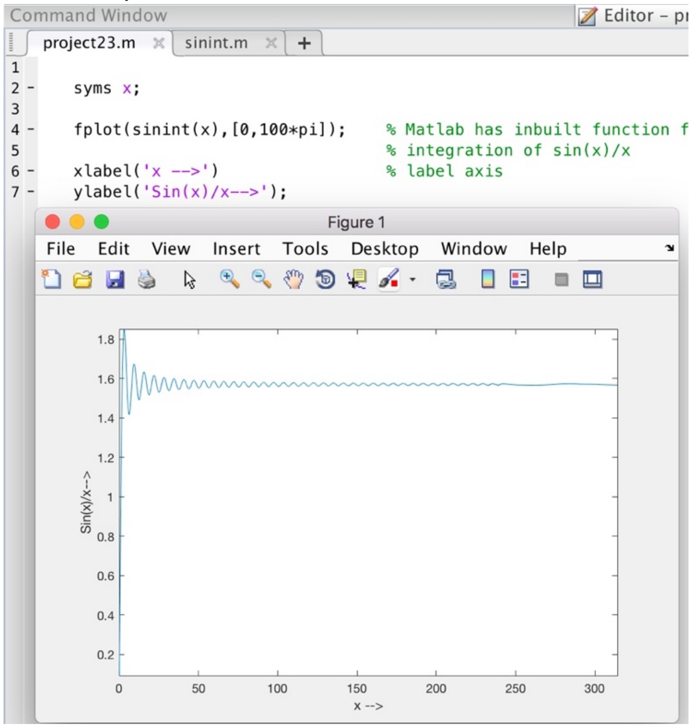
Integral of sin(x)/x limit from 0 to pi is 1.852 as we see in the graph below.



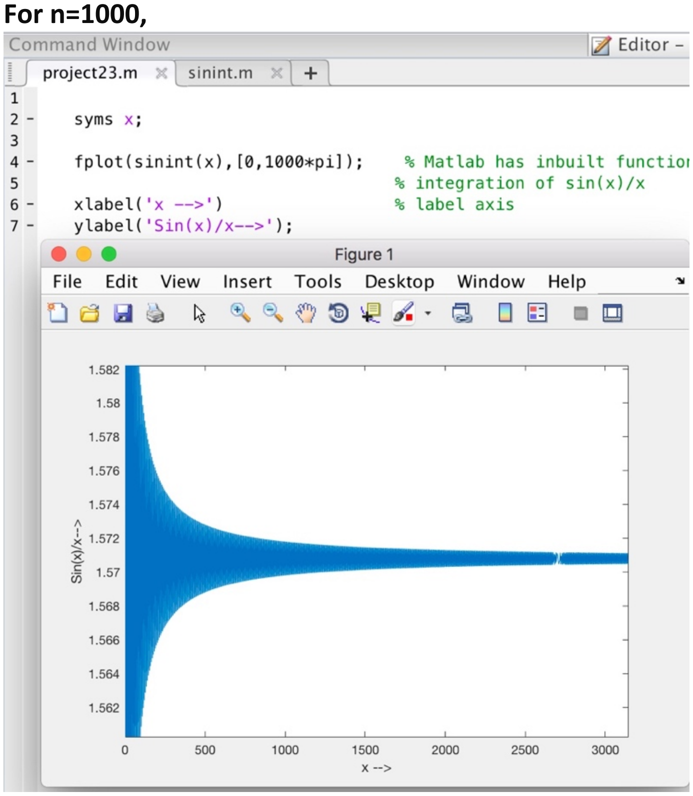
For n=10,



For n=100,



For n=1000,



Discussion of Results

We analysed the integrate sin(x)/x limit from 0 to pi for different values of n. We observe that as $n \to \infty$ the integral tends to zero. This integral is known as the Dirichlet integral.

v. References

- https://www.tutorialspoint.com/matlab/
- https://www.mathworks.com/help/matlab/ref/integral.html#btdd9x5
- Lecture 5 - Simple Monte-Carlo Methods (2up).pdf by professor Silvester

vi. Source Code

```
clc;
close all;
clear all;

syms x;

fplot(sinint(x),[0,1000*pi]);                %Matlab has inbuilt function for integral of sin(x)/x
xlabel('x -->')
ylabel('Sin(x)/x-->');

sample= 1000;

X=rand(1,sample);                            %generating random numbers
for j=1:sample
   Z(j)=sin(pi.*X(1,j))./X(1,j);    %storing computed value in XX
end
value1=sum(Z)/sample;                        %computed value by using random variables from MATLAB

syms y;
fplot(sinint(y), [0, pi]);
func=@(x)(sin(pi.*x)./x);
value2=integral(func,0,1);                   %computed value of function using matlab
X1=mean(X);
X2=std(X);
i1=X1-((1.96*X2)/sqrt(sample));              %confidence interval
i2=X1+((1.96*X2)/sqrt(sample));              %confidence interval
fprintf('The experimental value of integral is %d\n',value1);
fprintf('The theoretical value of integral is %d',value2);
```