EE 511 Simulation Methods for Stochastic Systems
Project #3: Investigations on Monte Carlo Methods (Chaitra Suresh -7434709345)

[Area-Estimation]

Goal:

i.) Generate n=500 samples (X, Y) of i.i.d 2-dimensional uniform random variables in the unit-square.
ii.) Write a function that counts how many of these samples fall within the region, D, formed by the intersection of two quarter unit-circles centered at the origin and at (1,1) as shown above.
iii.) Use these random samples to estimate the area of the inscribed region D. Use this area estimate to estimate the area of D. Do k=50 runs of these estimations. Plot the histogram of the k=50 area estimates.
iv.) Repeat the experiment with different numbers of uniform samples, n. Plot the sample variance of the Monte Carlo estimates as a function of your sample size n. Keep k=50 for all these runs. Comment on the sample variance of your estimates.

Algorithm / Routine:

Step 1: Generate n=500 samples (X,Y) of i.i.d 2- dimensional uniform random variables in unit square.

Step 2: Region D is formed by the intersection of two quarter unit circles centered at origin and at (1,1). The condition turns out as follows: D = { (x,y) : $x^2 + y^2 <= 1 \cap (x-1)^2 + (y-1)^2 <=1$ }

Step 3: Count the number of points out of n=500 samples which belongs to region D

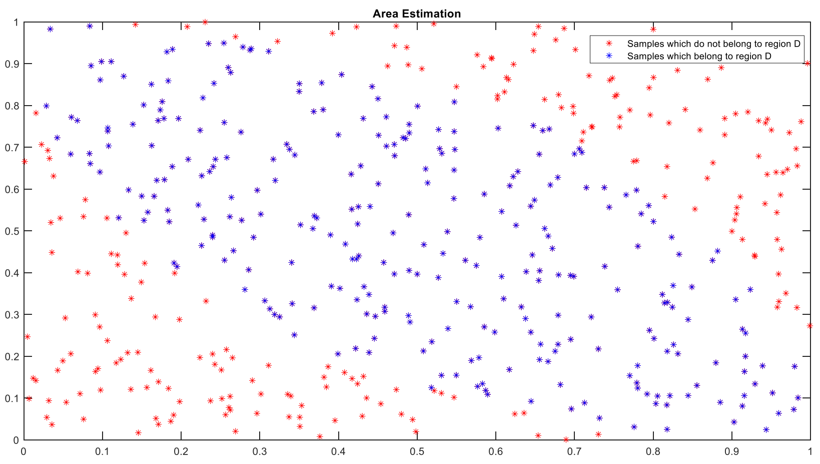
Step 4: Area of region D is estimated as $P((x,y)\epsilon D) = h = \frac{\text{\# number of samples in region D}}{\text{\# Total number of samples generated (n)}}$

Step 5: Repeat Step1~4 for k=50 run, estimate the sample variance as $\frac{1}{k}\sum_{i=1}^{k}(h(i) - \mu(h))^2$

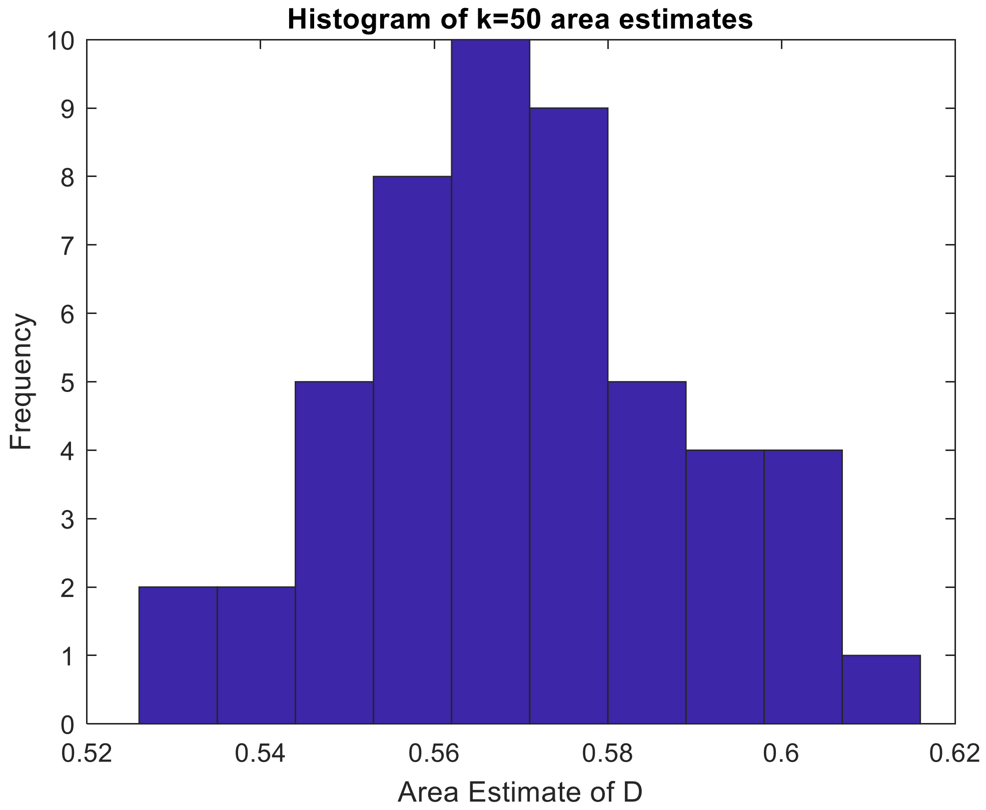
Step 6: Repeat the experiment with different number of uniform samples (n). Plot the sample variance of the Monte Carlo estimates as a function of your sample size n

Results:

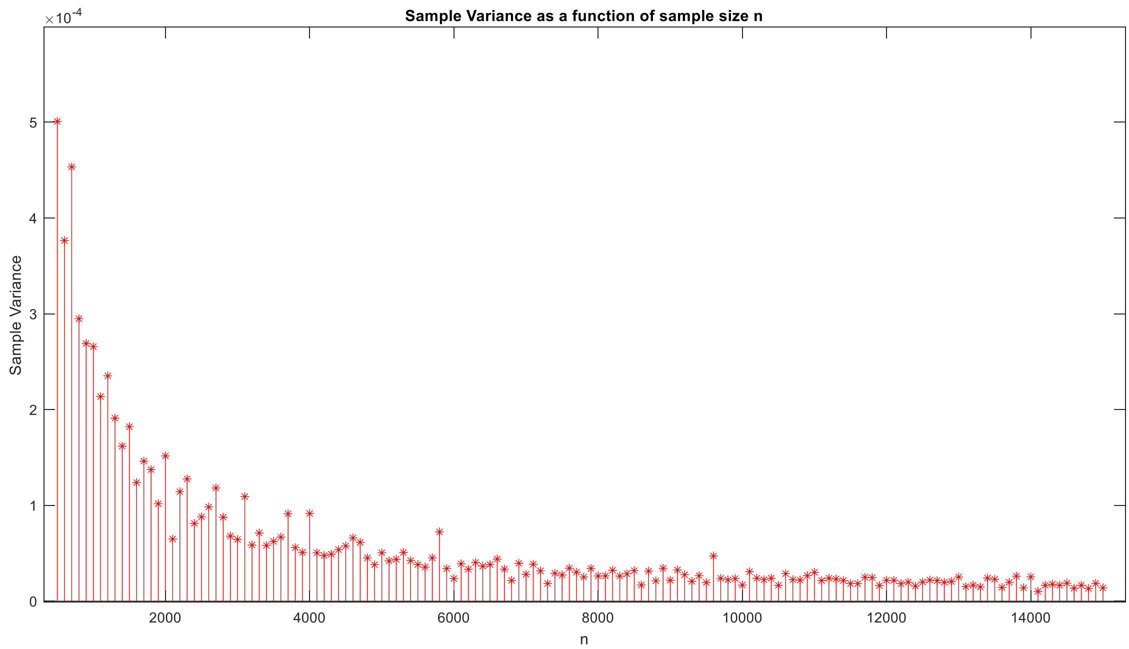
(i,ii)



(iii)



(iv)



Discussion:

1. Area Estimation for region D (Theoritical)

Area of quarter circle cenetered at origin + Area of quarter circle centered at (1,1) – Area of region D (overlapped twice) = Area of unit square

$$\frac{\pi}{4} + \frac{\pi}{4} \text{ - Area of region D } = 1*1$$

$$\text{Area of region D} = \frac{\pi}{4} + \frac{\pi}{4} - 1 = \frac{\pi}{2} - 1 = 1.5708 - 1 \cong 0.5708$$

From the result of (iii), it can be inferred from the histogram, that average area estimate for region D ~ 0.57

Thus, the area estimated by monte carlo simulation is in sync with theoretical area estimation

2. Comment on sample variance as a function of n:

From Law of Large numbers (LLN), as the sample size n approaches larger value, the monte carlo simulation reaches closer to the theoretical/ expected values.

$$I' = \frac{1}{n}\sum_{i=1}^{n} Id(x, y) = \frac{\#\text{ number of samples in region D}}{\#\text{ Total number of samples generated}} = \frac{\#\text{ number of samples in region D}}{n}$$

sample variance = $\frac{1}{k}\sum_{i=1}^{k}(h(i) - \mu(h))^2$

LLN : As n increases $h = P((x, y)\epsilon D)\ \xrightarrow{\text{mopd}}\ \mu(h)$

As n value approaches ∞ (larger value in practical scenario), I' (estimation of area through monte carlo simulation) approximates to a better area estimate of region D and h approximates to μ(h) (mean of h)

Thus the sample variance value decreases as sample size n increases, as the variation/ difference of area estimation decreases.

[Monte Carlo Integration and Variance Reduction Strategies]

Goal:

Use n=1000 random samples to obtain Monte Carlo estimates for the definite integrals:
(a) [1+sinh(2x)ln(x)]-1, x in [0.8,3]
(b) Exp[-x4 – y4], (x, y) in [-pi, pi]
Calculate the sample variance of the Monte Carlo estimates using a similar method as in problem 1.
Use the same number of random samples, n=1000, to obtain those Monte Carlo estimates. But this time incorporate stratification and importance sampling in the Monte Carlo estimation procedures. Compare the Monte Carlo estimates and their sample variances.
Discuss the quality of the Monte Carlo estimates from each method. Also discuss the strengths and weaknesses of stratification and importance sampling in Monte Carlo estimation.
Test your integral estimator on the following function with your own choice of n samples:
$f(x, y) = 20 + x^2 + y^2 - 10(cos[2\pi \times x] + cos[2\pi \times y])$ (x, y) in [-5, 5] for f(x,y)

Algorithm/ Routine:

(i) **Monte Carlo Integration Estimation**

Step 1: Generate 'n' uniform random samples in the integration interval of x / (x,y) [n=1000]

Step 2: Calculate mean of function value f(x) (or f(x,y) or integration estimate for 'n' values

$$h = \frac{\text{interval length}}{n}\sum_{x=1}^{n} f(x)$$

Step 3: Repeat Step 1~2 for k=50 times and estimate the integration (h) average over k

Step 4: Evaluate the variance of integration estimates for k different values obtained

$$\text{sample variance} = \frac{1}{k}\sum_{i=1}^{k}(h(i) - \mu(h))^2$$

(ii) **Monte Carlo Integration Estimation using Stratification**

Step 1: Divide the region of integration into N parts/ segments with different ‘n’ sample sizes depending on the nature of integration/curve. If the region of integration is over a constant sub region, we would require only 1 sample. If the region of integration has a slope, then 2-3 samples are sufficient for integration over the sub-region. If the region of integration has many distortions/curves, more samples are to be used to better approximate the integration.

Step 2: Generate ‘n_i ’ uniform random samples for each of the sub-divided interval of x / (x,y)

Step 3: Calculate mean of function value f(x) (or f(x,y) for interval (integration estimate) as

$$h = \sum_{i=1}^{N} \frac{\text{interval length*sum of function value for n samples}}{n}$$

Step 4: Repeat Step 1~3 for k=50 times and estimate the integration average over k

Step 5: Evaluate the variance of integration estimates for k different values obtained

$$\text{sample variance} = \frac{1}{k} \sum_{i=1}^{k} (h(i) - \mu(h))^2$$

(iii) **Monte Carlo Integration Estimation using Importance sampling**

Step 1: Choose an important sampling probability distribution function h(x) (or h(x,y))which is similar to the function f(x) (or f(x,y)) which has to be integrated.

Step 2: Generate ‘n’ random samples from the chosen important pdf in the integration interval of x / (x,y) [n=1000]

Step 3: Calculate mean of function value f(x) (or f(x,y) or integration estimates as for ‘n’ values as

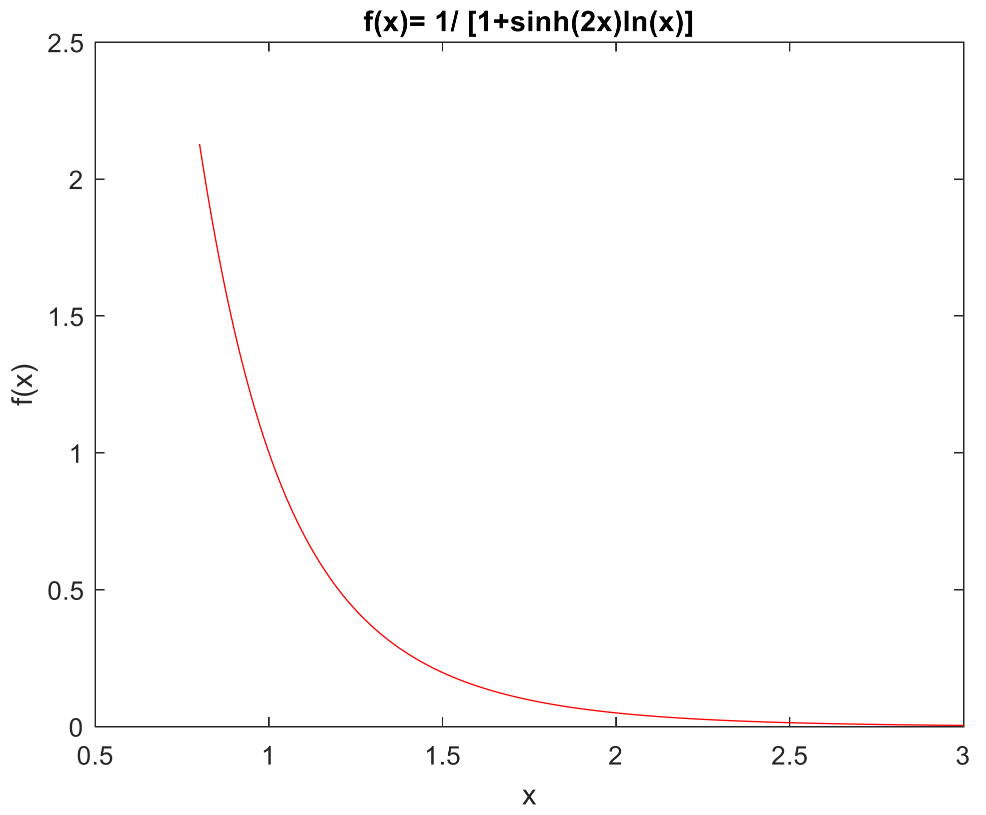
$$m = \frac{1}{n} \sum_{x=1}^{n} \frac{f(x)}{h(x)} \quad \text{or} \quad \frac{1}{n} \sum_{x=1}^{n} \sum_{y=1}^{n} \frac{f(x,y)}{h(x,y)}$$

Step 4: Repeat Step 1~3 for k=50 times and estimate the integration average over k

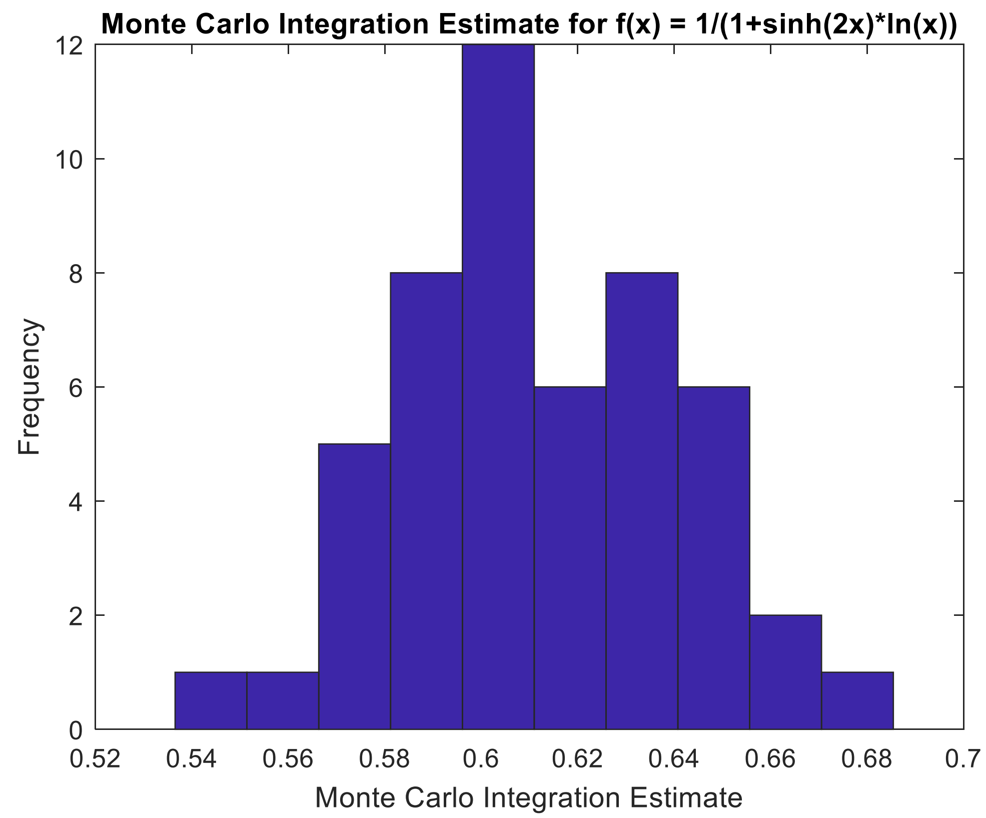
Step 5: Evaluate the variance of integration estimates for k different values obtained

$$\text{sample variance} = \frac{1}{k} \sum_{i=1}^{k} (m(i) - \mu(m))^2$$

Results:



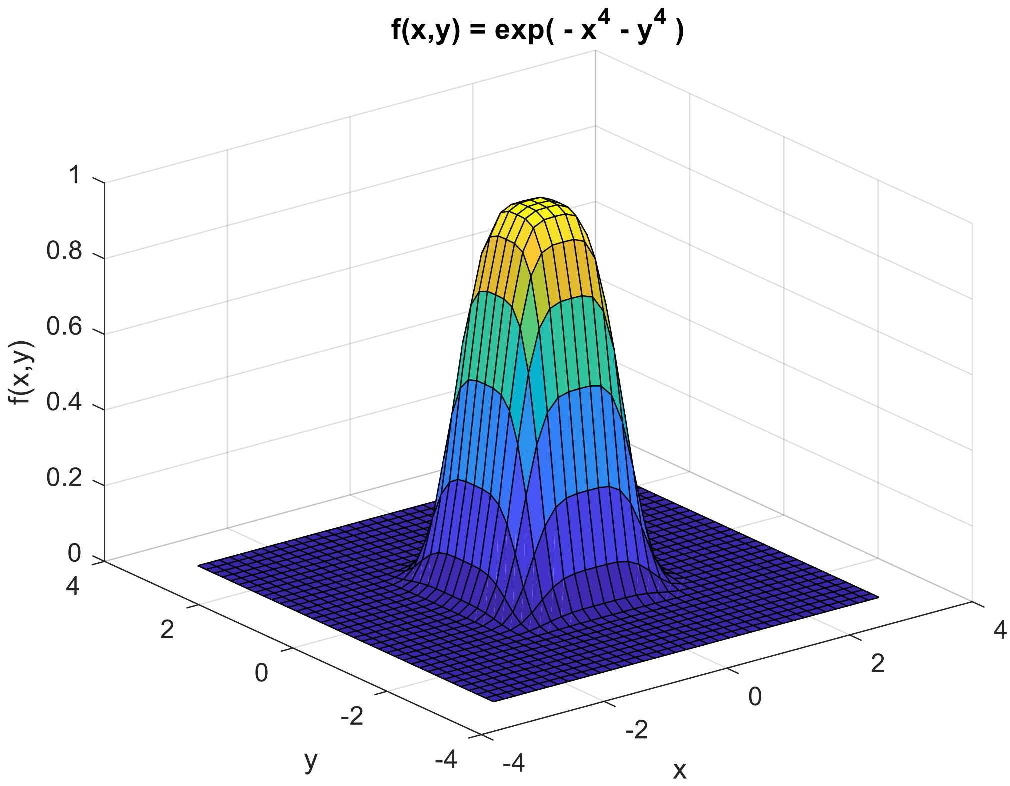
f(x)= 1/ [1+sinh(2x)ln(x)]
f(x)
x
2.5
2
1.5
1
0.5
0
0.5
1
1.5
2
2.5
3



Monte Carlo Integration Estimate for f(x) = 1/(1+sinh(2x)*ln(x))
Frequency
Monte Carlo Integration Estimate
12
10
8
6
4
2
0
0.52
0.54
0.56
0.58
0.6
0.62
0.64
0.66
0.68
0.7

Table 1: Integration estimate and variance for function $[1+\sinh(2x)\ln(x)]^{-1}$, x in [0.8,3]

Method	Integration Estimate	Variance
Monte Carlo Estimation	0.613978156149336	0.000794561236032052
Monte Carlo Estimation with stratification	0.6121	0.000148493855223465
Monte Carlo Estimation with importance sampling (Gaussian mean=0.5 standard deviation=0.8)	0.6011	0.0000867716492811518



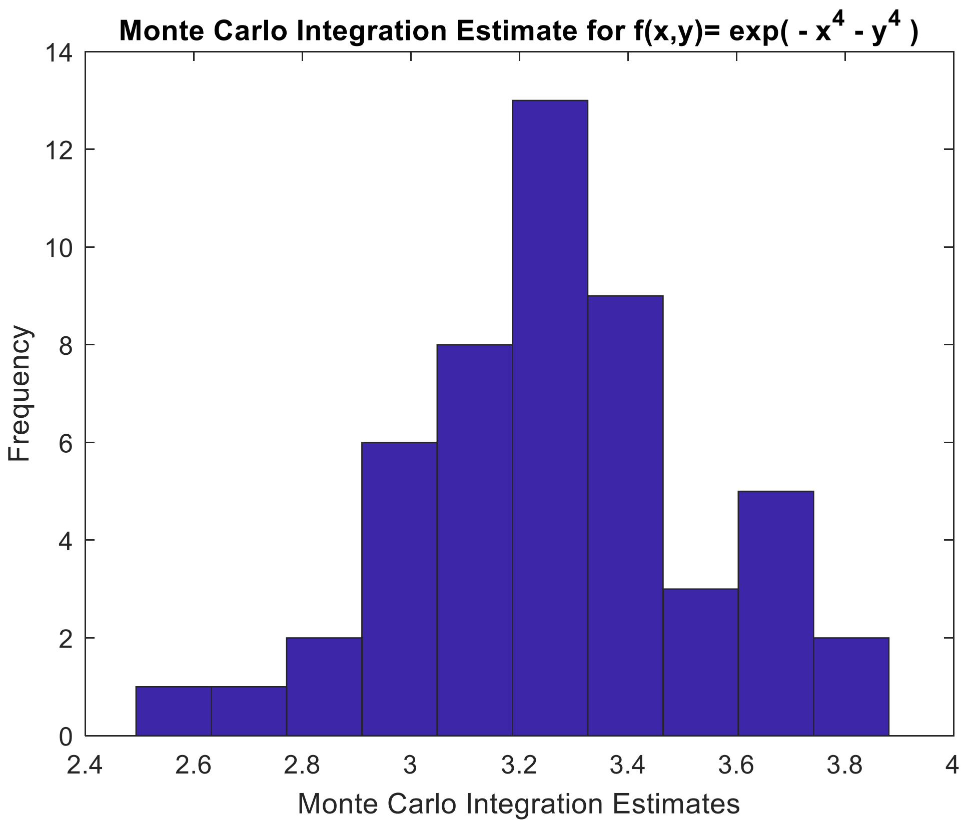
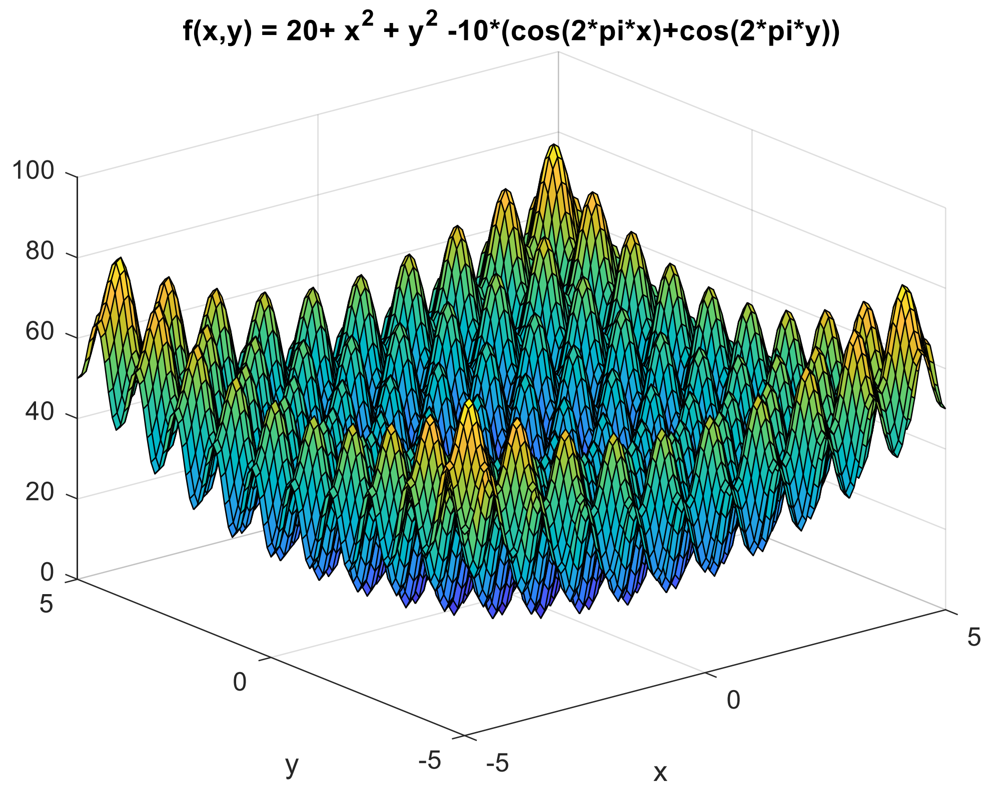


Table 2: Integration estimate and variance for function exp[-x4 – y4], (x, y) in [-pi, pi]

Method	Integration Estimate	Variance
Monte Carlo Estimation	3.26645769404764	0.0732102241813227
Monte Carlo Estimation with stratification	3.260664226077120	0.008489663597063
Monte Carlo Estimation importance sampling (Bivariate Gaussian mean=0 standard deviation=0.7)	3.264443377135568	0.003817638256676



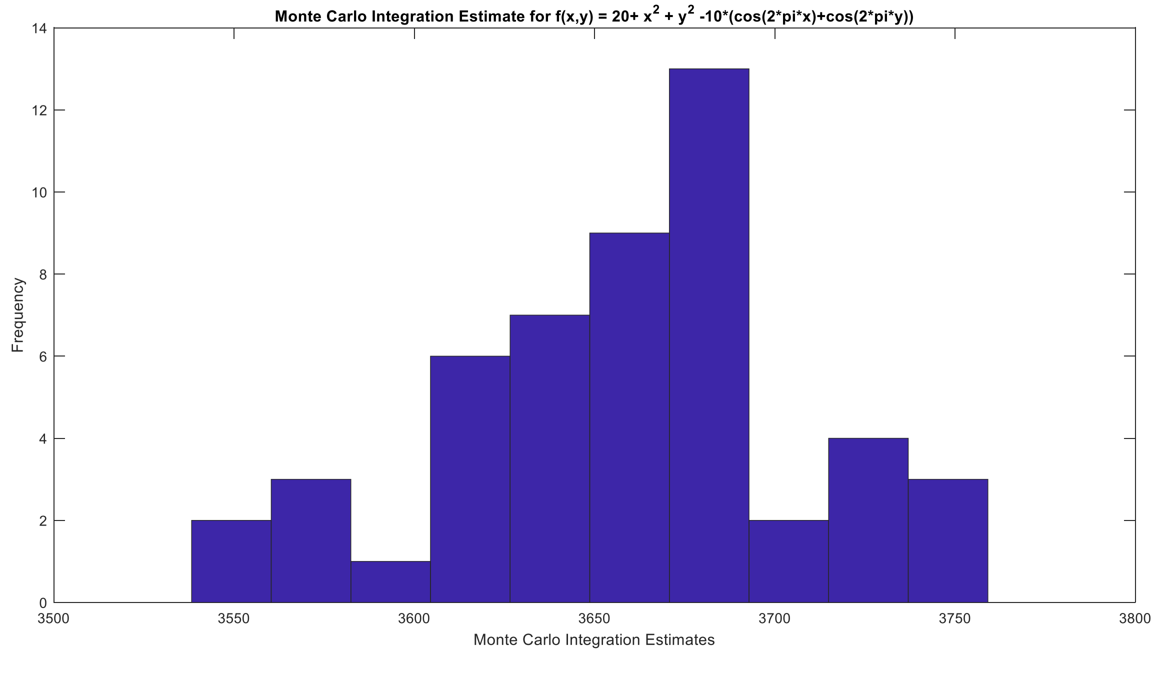


Table 3: Integration estimate and variance for function f(x,y)= 20+ x^2 +y^2 -10* (cos(2*pi*x) + cos (2 *pi*y)), (x, y) in [-5, 5]

Method	Integration Estimate	Variance
Monte Carlo Estimation	3660.19407509528	2485.95440453071
Monte Carlo Estimation with stratification	3461.331285068699	119.1995956242869

Discussion:

1. $[1+\sinh(2x)\ln(x)]^{-1}$, x in [0.8,3]

a. A simple Monte Carlo integration method for n samples resulted in integration estimate of 0.6139 and variance 0.0007945.
b. Monte Carlo Integration estimate using stratification, divided the interval as follows:
   0.8<=x<=1.2 with 500 samples (this region/interval has descent variation – more samples are required)
   1.2 < x<=2.5 with 350 samples
   2.5 < x <=3 with 150 samples (this region/interval has almost a constant function –less samples are required) Thus, using the above strategy, integration estimate of 0.6121 and variance 0.0001484 was obtained reducing the variance.
c. Monte Carlo Integration estimate using importance sampling:
   Gaussian probability density function with mean=0.5 and standard deviation = 0.8 is used as h(x) or importance pdf. The function was chosen since it is similar to the given function f(x).
   Thus, using the above strategy, integration estimate of 0.6011 and variance 0.00008677 was obtained reducing the variance.
d. Variance has been reduced significantly in both stratification and importance sampling. Variance is significantly reduced in importance sampling and the area estimate of importance sampling is more closer to the theoretical area 0.609553.

2. $\exp[-x^4 - y^4]$, (x, y) in [-pi, pi]

a. A simple Monte Carlo integration method for n=1000 samples resulted in integration estimate of 3.2664 and variance 0.07321.
b. Monte Carlo Integration estimate using stratification, divided into 5 intervals as follows:
   -1<=x,y<=1 with 800 samples (this region/interval has variation – more samples are required)
   -pi < x < -1 and -pi < y < pi with 50 samples
   1 < x< pi and -pi < y < pi with 50 samples
   1 < x< pi and -1 < y < 1 with 50 samples
   - pi < x< -1 and -1 < y < 1 with 50 samples
   Thus, using the above strategy, integration estimate of 3.2606and variance 0.008489 was obtained with reduced variance.
c. Monte Carlo Integration estimate using importance sampling:
   Gaussian probability density function with mean=0 and standard deviation = 0.7 is used as h(x) or importance pdf. The function was chosen since it is similar to the given function f(x).
   Thus, using the above strategy, integration estimate of 3.264443and variance 0.00381763 was obtained with reduced variance.
d. Variance has been reduced significantly in both stratification and importance sampling. Variance is significantly reduced in importance sampling and the area estimate of importance sampling is more closer to the theoretical area 3.28626.

3. $f(x,y)= 20+ x^2 +y^2 -10* (\cos(2*pi*x) + \cos (2 *pi*y))$, (x, y) in [-5, 5]

a. A simple Monte Carlo integration method for n=1000 samples resulted in integration estimate of 3660.194 and variance 2485.9544.
b. Monte Carlo Integration estimate using stratification, divided into 5 intervals as follows for n=9000
   -2<=x,y<=2 with 1000 samples (this region/interval has less variation – low samples are required)
   -5 < x < -2 and -5 < y < 5 with 3000 samples
   2 < x < 5 and -5 < y < 5 with 3000 samples

2 < x < 5 and -2 < y < 2 with 1000 samples

- 5 < x < -2 and -2 < y < 2 with 1000 samples

Thus, using the above strategy, integration estimate of 3461.331 and variance 119.1995 was obtained with reduced variance.

c. Variance has been reduced significantly in stratification. The area estimate of simple Monte Carlo estimation is closer to the theoretical area 3666.67.

4. a. Stratification in Monte Carlo estimation.

Strength:

1. Simple and useful method, dependent only on the data pattern, if it can be sub-divided into groups.
2. Reflects the population data statistics.

Weakness:

1. Tedious to sub-divide into groups
2. Cannot be applied to all data patterns (only can be divided into groups)

b. Importance sampling in Monte Carlo estimation.

Strength:

1. Reduced variance and more approximated area estimate
2. Efficiency is high (less randomness and more probabilistic approach)

Weakness:

1. To search for a similar distribution (h(x) – important sampling pdf) might be tedious and may not necessarily match each time.
2. Even if h(x)- importance sampling pdf is identified, it might be difficult to set the parameters to get good weights for evaluating integration estimate.

CODE:

1. Area Estimate [MATLAB]

```
clear all;

%%Part1 (i),(ii),(iii)
N=500;
 for j=1:50
x=rand(1,N);
y=rand(1,N);
count=0;
% ind=zeros(1,N);
for i=1:N
    a=x(1,i);
    b=y(1,i);
   if((((x(1,i)^2)+(y(1,i)^2))<=1)&& ((((x(1,i)-1)^2)+((y(1,i)-1)^2))<=1))
      D(1,i)=1;
      ind(1,count+1)=i;
      count=count+1;
   end
end
h(1,j)=count/N;
end
% figure(1);
% plot(x(1,:),y(1,:),'*r');
% hold on;
% plot(x(1,ind(1:count)),y(1,ind(1:count)),'*b');
% title('Area Estimation');

figure(2);
hist(h);
```

```
k=50;
var=(1/k)*sum((h-mean(h)).^2);

%%Part1 (iv)
l=1;
for n=500:100:15000
for j=1:50
x=rand(1,n);
y=rand(1,n);
count=0;
% ind=zeros(1,N);
for i=1:n
    a=x(1,i);
    b=y(1,i);
   if((((x(1,i)^2)+(y(1,i)^2))<=1)&& ((((x(1,i)-1)^2)+((y(1,i)-1)^2))<=1))
      D(1,i)=1;
      ind(1,count+1)=i;
      count=count+1;
   else
      D(1,i)=0;
   end
end
h(1,j)=count/n;
end
temp(1,l)=mean(h);
k=50;
var(1,l)=(1/k)*sum((h-mean(h)).^2);
l=l+1;
end

figure(3);
stem(500:100:15000,var,'*r');
```

2.

```
%function plot

n=1;
for x=0.8:0.0022:3
    f1(1,n)=1/(1+sinh(2*x)*log(x));
    f2(1,n)=5*(1/(0.7*sqrt(2*pi)))*exp(-((x-0.5).^2)/(2*(0.7.^2)));
    n=n+1;
end
figure(1);
plot(0.8:0.0022:3,f1,'r');
title('f(x) = 1/(1+sinh(2*x)*ln(x)) ')
m=1;
n=1;
for x=-pi:pi/18:pi
    n=1;
    for y=-pi:pi/18:pi
      f3(m,n)=exp(-(x.^4)-(y.^4));
      n=n+1;
    end
    m=m+1;
end
[X,Y]=meshgrid(-pi:pi/18:pi,-pi:pi/18:pi);
figure(2);
surf(X,Y,f3);
title('f(x,y) = exp( - x^4 - y^4 )')
%
m=1;
```

```
n=1;
for x=-5:0.1:5
    n=1;
    for y=-5:0.1:5
      f4(m,n)=20+(x.^2)+(y.^2)-10*(cos(2*pi*x)+cos(2*pi*y));
      n=n+1;
    end
    m=m+1;
end
[X,Y]=meshgrid(-5:0.1:5,-5:0.1:5);
figure(2);
surf(X,Y,f4);
title('f(x,y) = 20+ x^2 + y^2 -10*(cos(2*pi*x)+cos(2*pi*y))');

%integrall

without stratification and importance sampling
for j=1:50
N=1000;
x1=0.8.*ones(1,N)+(3-0.8).*rand(1,N);

x2=(-pi).*ones(1,N)+(2*pi).*rand(1,N);
y2=(-pi).*ones(1,N)+(2*pi).*rand(1,N);

x3=(-5).*ones(1,N)+(10).*rand(1,N);
y3=(-5).*ones(1,N)+(10).*rand(1,N);

integral1=0;
integral2=0;
integral3=0;
for i=1:N
   integral1=integral1+(1/(1+sinh(2*x1(1,i))*log(x1(1,i))));
   integral2=integral2+exp(-(x2(1,i).^4)-(y2(1,i).^4));
   integral3=integral3+(20+(x3(1,i).^2)+(y3(1,i).^2)-
10*(cos(2*pi*x3(1,i))+cos(2*pi*y3(1,i))));
end
h1(1,j)=((3-0.8)*integral1)/N;
h2(1,j)=((2*pi)*(2*pi)*integral2)/N;
h3(1,j)=(10*10*integral3)/N;
end
% figure(4);
% hist(h1);
%
% figure(5);
% hist(h2);
%
% figure(6);
% hist(h3);
estimate1=mean(h1)
estimate2=mean(h2)
estimate3=mean(h3)

k=50;
var1=(1/k)*sum((h1-mean(h1)).^2)
var2=(1/k)*sum((h2-mean(h2)).^2)
var3=(1/k)*sum((h3-mean(h3)).^2)


%using stratification
%function 1
```

```
for j=1:50
N=1000;
n1=500;
x1(1,1:n1)=0.8.*ones(1,n1)+(1.2-0.8).*rand(1,n1);
n2=350;
x1(1,n1+1:n1+n2)=1.2.*ones(1,n2)+(2.5-1.2).*rand(1,n2);
n3=150;
x1(1,n1+n2+1:n1+n2+n3)=2.5.*ones(1,n3)+(3-2.5).*rand(1,n3);
integral1=0;
integral2=0;
integral3=0;
for i=1:N
    if(i<=n1)
     integral1=integral1+(1/(1+sinh(2*x1(1,i))*log(x1(1,i))));
    elseif(i>n1 && i<=(n1+n2))
     integral2=integral2+(1/(1+sinh(2*x1(1,i))*log(x1(1,i))));
    else
     integral3=integral3+(1/(1+sinh(2*x1(1,i))*log(x1(1,i))));
    end
end
h1(1,j)=((1.2-0.8)*(integral1/n1)+(2.5-1.2)*(integral2/n2)+(3-
2.5)*(integral3/n3));
end
% figure(4);
% hist(h1);

estimate1=mean(h1)

k=50;
var1=(1/k)*sum((h1-mean(h1)).^2)
%
%%using importance sampling - function 1
%
for j=1:50
N=1000;
integral1=0;
count=0;
while(count<1000)
    b=random('Normal',0.5,0.8);
    if(b>=0.8 && b<=3)
       x1(1,count+1)=b;    %a(1,i);

integral1=integral1+((1/(1+sinh(2*x1(1,count+1))*log(x1(1,count+1))))/(6.3*(1/(
0.8*sqrt(2*pi)))*exp(-((x1(1,count+1)-0.5).^2)/(2*((0.8).^2)))));
       count=count+1;
    end
end
 h1(1,j)=(3-0.8)*(integral1/(count-1));
end
% figure(4);
% hist(h1);

estimate1=mean(h1)

k=50;
var1=(1/k)*sum((h1-mean(h1)).^2)

% function 2
%stratification

for j=1:50
```

```
N=1000;
n1=800;
x(1,1:n1)=-1.*ones(1,n1)+(2).*rand(1,n1);
y(1,1:n1)=-1.*ones(1,n1)+(2).*rand(1,n1);

n2=50;
x(1,n1+1:n1+n2)=-pi.*ones(1,n2)+(2*pi).*rand(1,n2);
y(1,n1+1:n1+n2)=pi.*ones(1,n2)+(1-pi).*rand(1,n2);

n3=50;
x(1,n1+n2+1:n1+n2+n3)=-pi.*ones(1,n3)+(2*pi).*rand(1,n3);
y(1,n1+n2+1:n1+n2+n3)=-pi.*ones(1,n3)+(-1+pi).*rand(1,n3);

n4=50;
x(1,n1+n2+n3+1:n1+n2+n3+n4)=pi.*ones(1,n4)+(1-pi).*rand(1,n4);
y(1,n1+n2+n3+1:n1+n2+n3+n4)=-1.*ones(1,n4)+(2).*rand(1,n4);

n5=50;
x(1,n1+n2+n3+n4+1:n1+n2+n3+n4+n5)=-pi.*ones(1,n5)+(-1+pi).*rand(1,n5);
y(1,n1+n2+n3+n4+1:n1+n2+n3+n4+n5)=-1.*ones(1,n5)+(2).*rand(1,n5);
integral1=0;
integral2=0;
integral3=0;
integral4=0;
integral5=0;
for i=1:N
    if(i<=n1)
     integral1=integral1+exp(-(x(1,i).^4)-(y(1,i).^4));
    elseif(i>n1 && i<=(n1+n2))
     integral2=integral2+exp(-(x(1,i).^4)-(y(1,i).^4));
    elseif(i>(n1+n2) && i<=(n1+n2+n3))
     integral3=integral3+exp(-(x(1,i).^4)-(y(1,i).^4));
    elseif(i>(n1+n2+n3) && i<=(n1+n2+n3+n4))
     integral4=integral4+exp(-(x(1,i).^4)-(y(1,i).^4));
    else
     integral5=integral5+exp(-(x(1,i).^4)-(y(1,i).^4));
    end
end
h1(1,j)=(4)*(integral1/n1)+(12)*(integral2/n2)+(12)*(integral3/n3)+(4)*(integra
l4/n4)+(4)*(integral5/n5);
end
% figure(4);
% hist(h1);

estimate1=mean(h1)

k=50;
var1=(1/k)*sum((h1-mean(h1)).^2)


%function 2
% importance sampling

for j=1:50
N=1000;
integral1=0;
count=0;
while(count<1000)
    b=random('Normal',0,0.7);
    c=random('Normal',0,0.7);
    if(b>=-pi && b<=pi && c>=-pi && c<=pi)
```

```
        x1(1,count+1)=b;    %a(1,i);
        y1(1,count+1)=c;
        integral1=integral1+((exp(-(x1(1,count+1).^4)-
(y1(1,count+1).^4)))/(24.25*(1/(0.7*0.7*(2*pi)))*exp(-((x1(1,count+1)).^2 -
((y1(1,count+1)).^2))/(2*((0.7).^2)))));
        count=count+1;
    end
end
 h1(1,j)=(2*pi*2*pi)*(integral1/(count-1));
end
% figure(4);
% hist(h1);

estimate1=mean(h1)

k=50;
var1=(1/k)*sum((h1-mean(h1)).^2)


%function3
%stratification

for j=1:50
% N=1000;
n1=1000;
x(1,1:n1)=-2.*ones(1,n1)+(4).*rand(1,n1);
y(1,1:n1)=-2.*ones(1,n1)+(4).*rand(1,n1);

n2=3000;
x(1,n1+1:n1+n2)=-5.*ones(1,n2)+(2*5).*rand(1,n2);
y(1,n1+1:n1+n2)=5.*ones(1,n2)+(2-5).*rand(1,n2);

n3=3000;
x(1,n1+n2+1:n1+n2+n3)=-5.*ones(1,n3)+(2*5).*rand(1,n3);
y(1,n1+n2+1:n1+n2+n3)=-5.*ones(1,n3)+(-2+5).*rand(1,n3);

n4=1000;
x(1,n1+n2+n3+1:n1+n2+n3+n4)=5.*ones(1,n4)+(2-5).*rand(1,n4);
y(1,n1+n2+n3+1:n1+n2+n3+n4)=-2.*ones(1,n4)+(4).*rand(1,n4);

n5=1000;
x(1,n1+n2+n3+n4+1:n1+n2+n3+n4+n5)=-5.*ones(1,n5)+(-2+5).*rand(1,n5);
y(1,n1+n2+n3+n4+1:n1+n2+n3+n4+n5)=-2.*ones(1,n5)+(4).*rand(1,n5);
integral1=0;
integral2=0;
integral3=0;
integral4=0;
integral5=0;
for i=1:n1+n2+n3+n4+n5
    if(i<=n1)
     integral1=integral1+(20+(x(1,i).^2)+(y(1,i).^2)-
10*(cos(2*pi*x(1,i))+cos(2*pi*y(1,i))));
    elseif(i>n1 && i<=(n1+n2))
     integral2=integral2+(20+(x(1,i).^2)+(y(1,i).^2)-
10*(cos(2*pi*x(1,i))+cos(2*pi*y(1,i))));
    elseif(i>(n1+n2) && i<=(n1+n2+n3))
     integral3=integral3+(20+(x(1,i).^2)+(y(1,i).^2)-
10*(cos(2*pi*x(1,i))+cos(2*pi*y(1,i))));
    elseif(i>(n1+n2+n3) && i<=(n1+n2+n3+n4))
     integral4=integral4+(20+(x(1,i).^2)+(y(1,i).^2)-
10*(cos(2*pi*x(1,i))+cos(2*pi*y(1,i))));
```

```
    else
     integral5=integral5+(20+(x(1,i).^2)+(y(1,i).^2)-
10*(cos(2*pi*x(1,i))+cos(2*pi*y(1,i))));
    end
end
h1(1,j)=(16)*(integral1/n1)+(30)*(integral2/n2)+(30)*(integral3/n3)+(9)*(integr
al4/n4)+(9)*(integral5/n5);
end
% figure(4);
% hist(h1);

estimate1=mean(h1)

k=50;
var1=(1/k)*sum((h1-mean(h1)).^2)
```