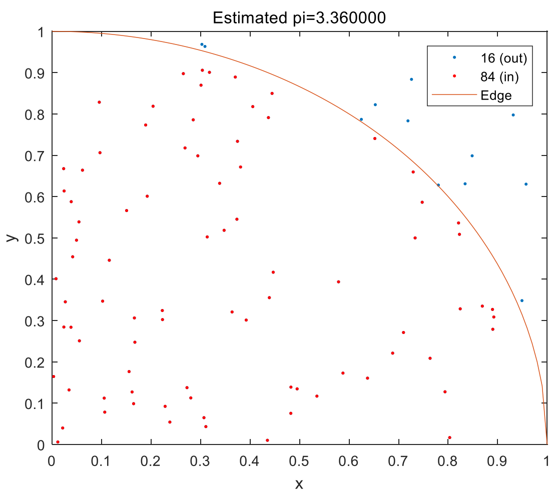
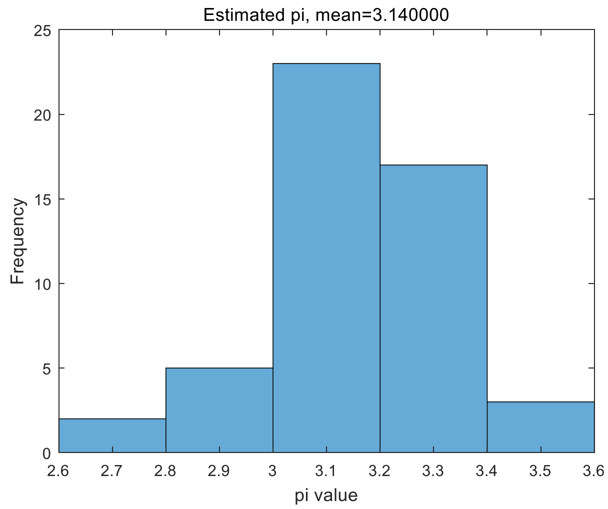
EE511 Project#4 Monte Carlo Method

This project is implemented using MATLAB. Codes are attached in the end of this document and also available at https://github.com/uscwy/ee511project4/blob/master/project4.m

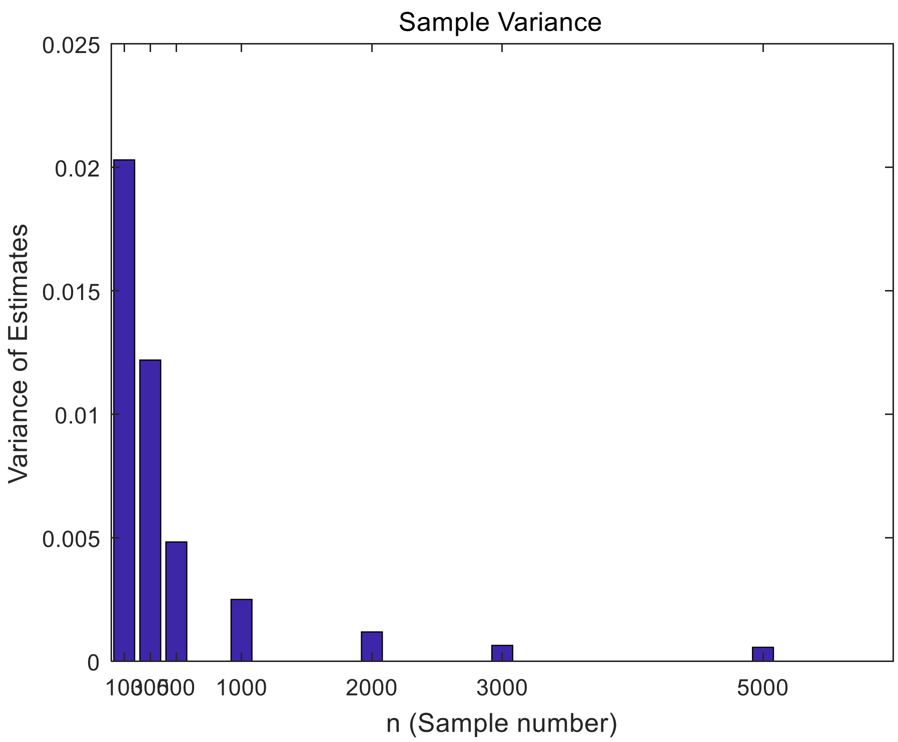
[Pi-Estimation]

By generating 2-D uniform points and counting the points fall within circle, the pi value could be estimated using equation pi=4(points within circle/total points). This left plot shows the estimation process of n=100 points. The right plot is the histogram of 50 times of estimation.





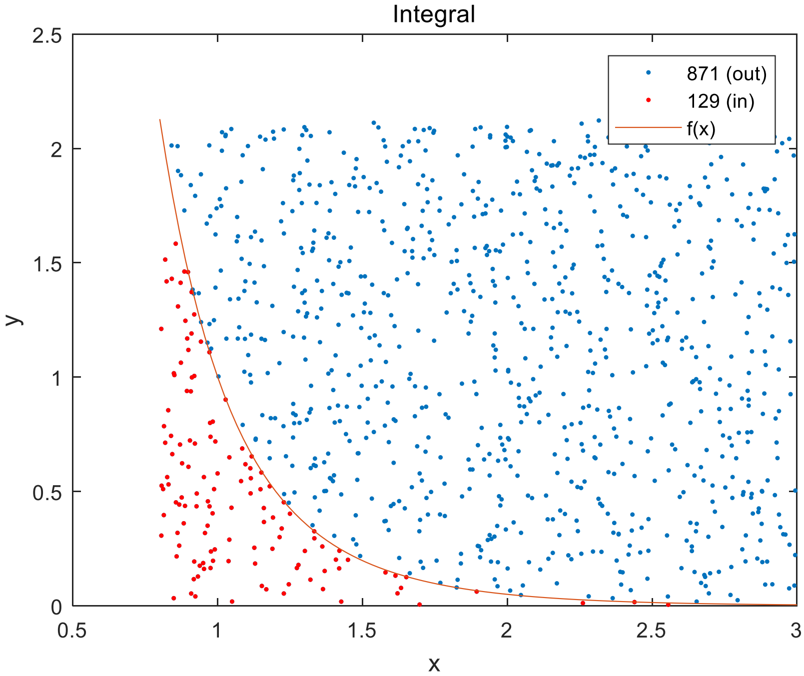
Then by generating different number of sample points (n=100, 200, 500, 1000, 3000, 5000), we could find the variance goes smaller as n increased. The below bar plot shows the relationship between sample number and variance of estimation.



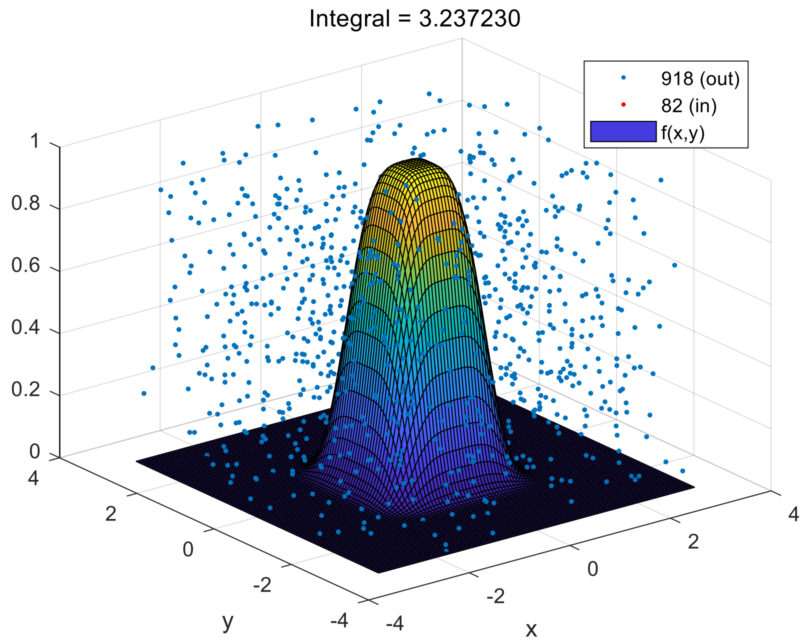
[Monte Carlo Integration and Variance Reduction Strategies]

The first strategy is to estimate integral by generate uniform random points and counting points fall in the integral area. Since this is a monotonic function, the sampling space could be easily calculated as the rectangular area.

Then the estimated integral = the area of rectangular * (points below f(x)/total points). The below plot shows the estimation process of (a) $[1+\sinh(2x)\ln(x)]^{-1}$, x in [0.8,3]



The second is multi-dimensional integral. Before generating uniform sampling points in 3-D space, we need to determine the height of sampling space. By observing the expression of f(x,y), the height of sample space is equal to the maximum value of f(x,y) which is f(0,0). The below plot shows the estimation process.



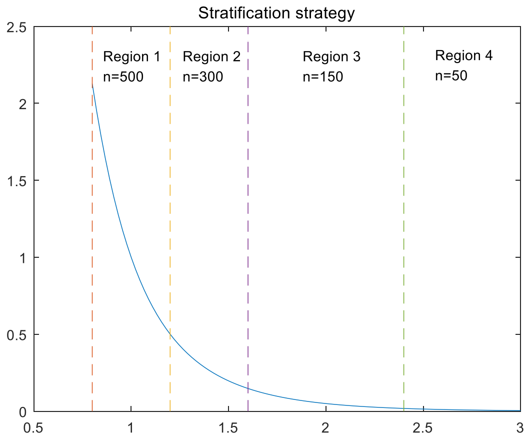
Then keeping the n=1000 samples for k=50 runs, variance and means are shows as following table.

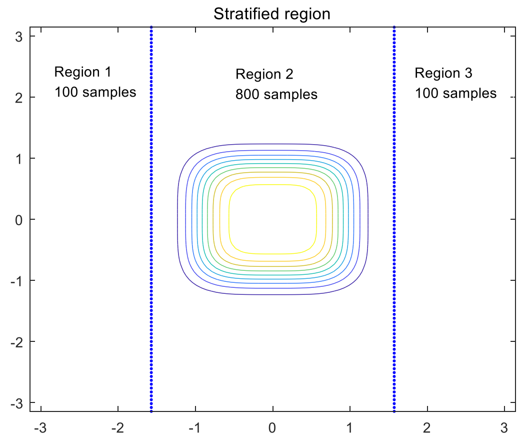
Table 1 Variance and Mean of Uniform Sample Points

Integrand	Mean Integral	Variance	Number of Samples
(a) $[1+\sinh(2x)\ln(x)]^{-1}$,	0.590060	0.002088	1000
(b) $\mathrm{Exp}[-x^4 - y^4]$,	3.273550	0.116531	1000

Stratification sampling:

The below plot shows my stratification strategies for the two integrands. Keeping total number of sample unchanged, I assigned more samples to the region where the variance of integrand is larger.





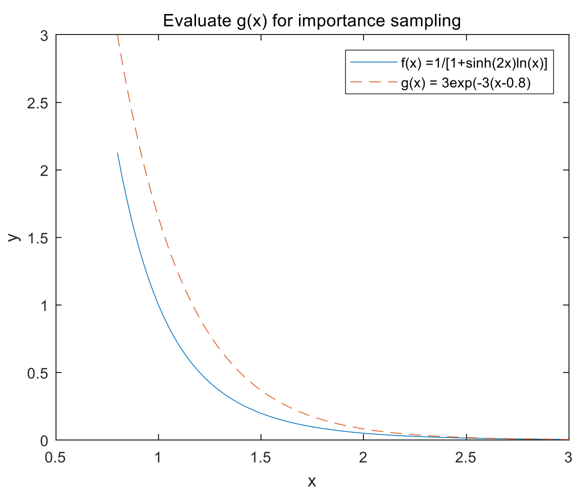
According to the formula, $I = \frac{1}{N}\sum_{i=0}^{N} f(xi, yi)\Delta x\Delta y$, in which $\Delta x, \Delta y$ are the intervals of region where xi or yi belong to, the integral could be estimated within each region and then sum them up to get the result. The below table shows the result of variance and mean of k=50 runs.

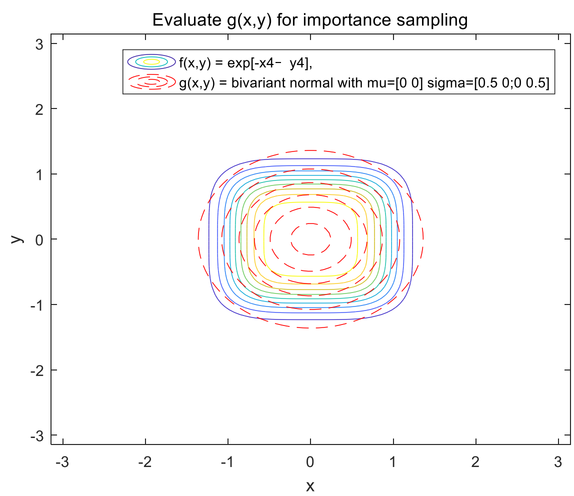
Table 2 Variance and Mean by Stratification method

Integrand	Mean Integral	Variance	Number of Samples
(a) $[1+\sinh(2x)\ln(x)]^{-1}$,	0.609057	0.000081	1000
(b) $\mathrm{Exp}[-x^4 - y^4]$,	3.284085	0.040741	1000

Importance sampling

By observing the two integrands, the first looks like an exponential pdf and the seconds looks like a bivariant normal pdf. So we can choose g(x) = exppdf(x-0.8, 1/3) to determine the distribution of samples for the first integrand. And choose g(x,y)=mvnpdf(x,y) with mu=[0 0], Sigma=$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$ to determine the distribution of samples for the second integrand. The below plot shows the relationship between g(x,y) and f(x,y).





According to the formula, $I = \frac{1}{N}\sum_{i=0}^{N} \frac{f(xi,yi)}{g(xi,yi)}$ the integral could be estimated as the following table.

Table 3 Variance and Mean by Importance sampling

Integrand	Mean Integral	Variance	Number of Samples
(a) $[1+\sinh(2x)\ln(x)]^{-1}$,	0.611681	0.000005	1000
(b) $\mathrm{Exp}[-x^4 - y^4]$,	3.281630	0.001787	1000

Comparation of different sampling method:

Table 4 Comparation of different MC methods. k=50 n=1000

Integrand	Uniform sampling	Stratification sampling	Importance sampling
(a) $[1+\sinh(2x)\ln(x)]^{-1}$,	M=0.590060 V=0.002088	M=0.609057 V=0.000081	M=0.611681 V=0.000005
(b) $\mathrm{Exp}[-x^4 - y^4]$,	M=3.273550 V=0.116531	M=3.284085 V=0.040741	M=3.281630 V=0.001787

By showing the mean and variance in one table, we can find that uniform sampling has the biggest variance, stratification sampling can reduce variance significantly; the importance sampling can reduce variance further.

The strengths and weakness

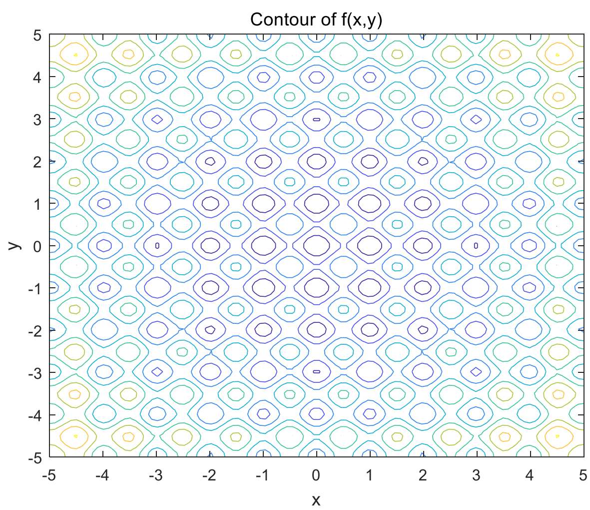
Uniform sampling is the simple way to estimate integral and can be used to estimate most integral. The variance could be reduced by increasing number of samples. But the weakness is that the variance is the biggest among the three methods. To get higher precision, we need a very large n and a lot of time of calculation.

Stratification sampling can reduce variance significantly. But we need to analysis the integrand and draw its plot to find proper regions. This method depends on the integrand. In multi-dimensional space, it may very difficult to observe and stratify the integrand.

Importance sampling can reduce variance further than stratification. But this method depends on whether we can find a proper pdf which is similar to the integrand and easy to generate random samples.

Test integral estimator

The last function is very complex (as plot show) and hard to find a proper pdf or stratified region. I use uniform stratification sampling method which splits the whole x-y plane uniformly into d*d small pieces. In each piece, we can generate n random samples to estimate the integral using the same formula as stratification sampling method. The total number of sample is equal to d*d*n.



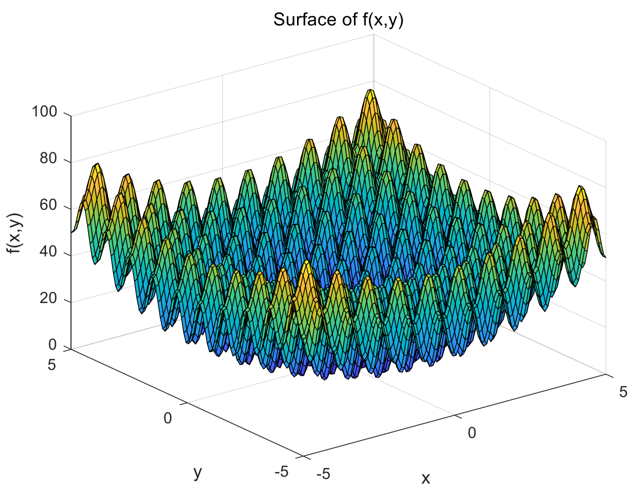


Table 5 Integral result

Number of samples (d*d*n)	Integral (mean)	Variance
99*99*10	3666.700429	0.432426
99*99*100	3666.733794	0.057561
99*99*1000	3666.667731	0.003142

```
%EE511 Project 4
%Author: Yong Wang <yongw@usc.edu>
n=100;
X = rand([n 1]);
Y = rand([n 1]);
figure;
plot(X,Y,'.');
hold on;
A = X.^2 + Y.^2;
plot(X(A<=1),Y(A<=1),'r.');
x=linspace(0,1);
y = sqrt(1 - x.^2);
plot(x,y);
legend(sprintf('%d (out)',length(find(A>1))),sprintf('%d
(in)',length(find(A<=1))),'Edge','Location','NE');
xlabel("x");
ylabel('y');
epi = 4 * length(find(A<=1))/n;
title(sprintf('Estimated pi=%f',epi));
k=50;
P=zeros(k,1);
for i=1:k
    X = rand([n 1]);
    Y = rand([n 1]);
    A = X.^2 + Y.^2;
    P(i,1)= 4 * length(find(A<1))/n;
end
figure;
histogram(P);
title(sprintf('Estimated pi, mean=%f',sum(P)/length(P)));
xlabel("pi value");
ylabel('Frequency');
%% Variance
N = [100 300 500 1000 2000 3000 5000];
V = zeros(length(N),1);
for j = 1:length(N)
    for i=1:k
        X = rand([N(j) 1]);
        Y = rand([N(j) 1]);
        A = X.^2 + Y.^2;
        P(i,1)= 4 * length(find(A<1))/N(j);
    end
    V(j) = var(P);
    fprintf("n=%d mean=%f, var=%f\n",N(j), sum(P)/length(P),V(j));
end
figure;
bar(N,V);
title('Sample Variance');
xlabel("n (Sample number)");
```

```
ylabel('Variance of Estimates');

%% Monte Carlo integral by generating uniform points
n=1000;
x1=0.8;
x2=3.0;
y1 = 1/(1+sinh(x1*2).*log(x1));
y2 = 1/(1+sinh(x2*2).*log(x2));
X=(x2-x1).*rand(n,1)+x1;
Y=(y2-y1).*rand(n,1)+y1;
A=1./(1+sinh(X*2).*log(X));
figure;
plot(X,Y,'.');
hold on;
plot(X(A>Y),Y(A>Y),'r.');
x=linspace(0.8,3);
y=1./(1+sinh(x*2).*log(x));
plot(x,y);
legend(sprintf('%d (out)',length(find(A<Y))),sprintf('%d
(in)',length(find(A>Y))),'f(x)','Location','NE');
xlabel('x');
ylabel('y');
title(sprintf('Integral = %d',abs(x2-x1)*abs(y2-y1)*length(find(A>Y))/n));
N=[1000];
V=zeros(length(N),1);
for j=1:length(N)
    int = zeros(k,1);
    for i=1:k
        n=N(j);
        x1=0.8;
        x2=3.0;
        y1 = 1/(1+sinh(x1*2).*log(x1));
        y2 = 1/(1+sinh(x2*2).*log(x2));
        X=(x2-x1).*rand(n,1)+x1;
        Y=(y2-y1).*rand(n,1)+y1;
        A=1./(1+sinh(X*2).*log(X));
        int(i,1) = abs(x2-x1)*abs(y2-y1)*length(find(A>Y))/n;
    end
    V(j) = var(int);
    fprintf("1: Mean=%f, Variance=%f\n",sum(int)/k, V(j));
end

n=1000;
x1=-pi;
x2=pi;
y1 = -pi;
y2 = pi;
z1 = exp(-1*0^4 - 0^4);
z2 = exp(-1*pi^4 - pi^4);
```

```
X=(x2-x1).*rand(n,1)+x1;
Y=(y2-y1).*rand(n,1)+y1;
Z=(z2-z1).*rand(n,1)+z1;
A=exp(-1*(X.^4) - Y.^4);
figure;
scatter3(X,Y,Z,'.');
hold on;
scatter3(X(A>Z),Y(A>Z),Z(A>Z),'r.');
x=linspace(x1,x2);
y=linspace(y1,y2);
surf(x,y,exp(-1*(x.^4) - y'.^4));
legend(sprintf('%d (out)',length(find(A<Z))),sprintf('%d
(in)',length(find(A>Z))),'f(x,y)','Location','NE');
xlabel('x');
ylabel('y');
int = abs(x2-x1)*abs(y2-y1)*abs(z2-z1)*length(find(A>Z))/n;
title(sprintf('Integral = %f',int));
figure;
contour(x,y,exp(-1*(x.^4) - y'.^4));
hold on;
plot(-pi/2,y,'b.',pi/2,y,'b.');
title('Stratified region');
N=[1000];
V=zeros(length(N),1);
for j=1:length(N)
    int = zeros(k,1);
    for i=1:k
        n=N(j);
        X=(x2-x1).*rand(n,1)+x1;
        Y=(y2-y1).*rand(n,1)+y1;
        Z=(z2-z1).*rand(n,1)+z1;
        A=exp(-1*X.^4 - Y.^4);
        int(i,1) = abs(x2-x1)*abs(y2-y1)*abs(z2-z1)*length(find(A>Z))/n;
    end
    V(j) = var(int);
    fprintf("2: Mean=%f, Variance=%f\n",sum(int)/k, V(j));
end
%% Stratification
v = [0.8 1.2;1.2 1.6;1.6 2.4;2.4 3.0];
s = [500 300 150 50];
k=50;
int = zeros(k,1);
for i=1:k
    r = 0;
    for j=1:length(s)
        n = s(j);
        x1=v(j,1); x2=v(j,2);
        x=(x2-x1).*rand(n,1)+x1;
        y=1./(1+sinh(x*2).*log(x));
```

```
        r = r + sum(y.*(x2-x1))/n;
        if j == 1
            tt=y;
        end
    end
    int(i) = r;
end
fprintf("Stratification 1: integral=%f, var=%f\n", sum(int)/k, var(int));
v = [-pi -pi/2;-pi/2 pi/2;pi/2 pi];
s = [100 800 100];
k=50;
int = zeros(k,1);
for i=1:k
    r = 0;
    for j=1:length(s)
        n = s(j);
        x1=v(j,1); x2=v(j,2);
        y1=-pi;y2=pi;
        x=(x2-x1).*rand(n,1)+x1;
        y=(y2-y1).*rand(n,1)+y1;
        z=exp(-1*x.^4 - y.^4);
        r = r + sum(z.*(x2-x1)*(y2-y1))/n;
        if j == 1
            tt=z;
        end
    end
    int(i) = r;
end
fprintf("Stratification 2: integral=%f, var=%f\n", sum(int)/k, var(int));
%% Importance sampling
n=1000;
for i=1:k
    x=exprnd(1/3,n,1)+0.8;
    f=1./(1+sinh(x*2).*log(x));
    g=exppdf(x-0.8,1/3);
    int(i)=sum(f./g)/n;
end
fprintf("Importance sampling 1: integral=%f, var=%f\n", sum(int)/k, var(int));
n=1000;
mu=[0 0];
sigma=[0.5 0;0 0.5];
for i=1:k
    x=mvnrnd(mu,sigma,n);
    f=exp(-1*x(:,1).^4 - x(:,2).^4);
    g=mvnpdf(x,mu,sigma);
    int(i)=sum(f./g)/n;
end
fprintf("Importance sampling 2: integral=%f, var=%f\n", sum(int)/k, var(int));
%% last integral
```

```
x=linspace(-5,5);
y=linspace(-5,5);
[X,Y] = meshgrid(x,y);
Z=20+X.^2+Y.^2-10*(cos(2*pi*X)+cos(2*pi*Y));
k=50;
n=10;
int=zeros(k,1);
for t=1:k
    r=0;
    for i=2:size(X,1)
        for j=2:size(Y,2)
            dx = X(i,j)-X(i,j-1);
            dy = Y(i,j)-Y(i-1,j);
            x=dx*rand(n,1)+X(i,j-1);
            y=dy*rand(n,1)+Y(i-1,j);
            z=20+x.^2+y.^2-10*(cos(2*pi*x)+cos(2*pi*y));
            r = r + sum(z*dx*dy)/n;
            %fprintf("dx=%f dy=%f, %f-%f %f-%f %d %d\n",dx,dy,X(i,j),X(i-
1,j),Y(i,j),Y(i,j-1),i,j);
        end
    end
    fprintf("Integral=%f\n",r);
    int(t)=r;
end
fprintf("Integral=%f Var=%f n=%d\n",sum(int)/k,var(int),n);
%% draw some plot for report document
figure;
x=linspace(0.8,3);
y=1./(1+sinh(x*2).*log(x));
plot(x,y);
hold on;
y=exppdf(x-0.8,1/3);
plot(x,y,'--');
title('Evaluate g(x) for importance sampling');
x=linspace(-pi,pi);
y=linspace(-pi,pi);
figure;
contour(x,y,exp(-1*(x.^4) - y'.^4));
title('Evaluate g(x,y) for importance sampling');
hold on;
mu = [0 0];
SIGMA = [0.5 0; 0 0.5];
[X,Y] = meshgrid(x,y);
z=mvnpdf([X(:) Y(:)], mu, SIGMA);
z=reshape(z,length(x),length(y));
contour(x,y,z,'r--');
figure;
x=linspace(-5,5);
y=linspace(-5,5);
```

```
[X,Y] = meshgrid(x,y);
Z=20+X.^2+Y.^2-10*(cos(2*pi*X)+cos(2*pi*Y));
surf(x,y,Z);
```