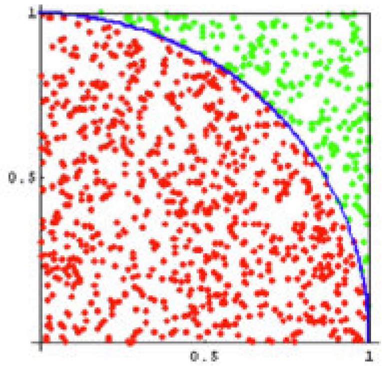
Excercise 1

Question Description

The Monte Carlo method can be used to generate an approximate value of pi. The figure below shows a unit square with a quarter of a circle inscribed. The area of the square is 1 and the area of the quarter circle is Pi/4. Write a script to generate random points that are distributed uniformly in the unit square. The ratio between the number of points that fall inside the circle (red points) and the total number of points thrown (red and green points) gives an approximation to the value of pi/4. This process is a Monte Carlo simulation approximating pi. Let N be the total number of points thrown. When N=50, 100, 200, 300, 500, 1000, 5000, what are the estimated pi values, respectively? For each N, repeat the throwing process 100 times, and report the mean and variance. Record the means and the corresponding variances in a table.



Implementation Details

- In the first part, trying to have a clear overview of the whole method, we use [matplotlib (https://www.baidu.com/link?url=AkPiPYbYiYO0fQwZi0pVbV7UvxFba4Bn6uHBHPOWsim&wd=&eqid=d5c1106300015827000000065c](https://www.baidu.com/link?url=AkPiPYbYiYO0fQwZi0pVbV7UvxFba4Bn6uHBHPOWsim&wd=&eqid=d5c1106300015827000000065c) to draw the random distribution of N observations in the given area. Images are slowly shown, so we have to try a limit amount of samples in this part.
- In order to further explore this state of method and give the analysis of the experiment results, we increase the times of repetitive experiments and use a huge amount of observations. Luckily, the experiment is well-conduct and show results as we expected. N = [5,10,20,50,100,500,1000,5000] correspondingly.

Monte Carlo

In [54]:

```
import random
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def drawCir(a,b,r,plot_num):
    theta = np.arange(0,(1/2)*np.pi,0.01)
    x = a + r*np.cos(theta)
    y = b + r*np.sin(theta)
#     plot_num.plot([x],[y],'b.')
#     plot_num.axis('equal')


def calPai(n,plot_num):
    r = 1.0
    a,b = (0.0,0.0)
    x_pos = a+r
    y_pos = b+r

    count = 0
    for i in range(0,n):
        x = random.uniform(0,x_pos)
        y = random.uniform(0,y_pos)
        if x*x + y*y <= 1.0:
            count += 1
#             plot_num.plot([x],[y],'r.')
            pass
        else:
#             plot_num.plot([x],[y],color='#40fd14',marker='.')
            pass

    pi = (count/float(n))*4
#     plt.show()
    return pi
```

Experiment Results

In [57]:

```
# plot1
print("\n-------------------------------------------------------\n")
n = 5
a,b,r = (0.,0.,1.)
# plt.figure(figsize=(15,15))
# plot1 = plt.subplot(4,2,1)
# drawCir(a,b,r,plot1)
pi1 = []
for i in range (100):
    pi1.append(calPai(n,plot1))
print("The value of Pi is : "+str(pi1)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(pi1))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(pi1))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(pi1))+" .\n")

# plot2
print("\n-------------------------------------------------------\n")
n = 10
a,b,r = (0.,0.,1.)
# plt.figure(figsize=(15,15))
# plot2 = plt.subplot(4,2,2)
# drawCir(a,b,r,plot2)
pi2 = []
for i in range (100):
    pi2.append(calPai(n,plot2))
print("The value of Pi is : "+str(pi2)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(pi2))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(pi2))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(pi2))+" .\n")

# plot3
print("\n-------------------------------------------------------\n")
n = 20
a,b,r = (0.,0.,1.)
# plt.figure(figsize=(15,15))
# plot3 = plt.subplot(4,2,3)
# drawCir(a,b,r,plot3)
pi3 = []
for i in range (100):
    pi3.append(calPai(n,plot3))
print("The value of Pi is : "+str(pi3)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(pi3))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(pi3))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(pi3))+" .\n")

# plot4
print("\n-------------------------------------------------------\n")
n = 50
a,b,r = (0.,0.,1.)
# plt.figure(figsize=(15,15))
# plot4 = plt.subplot(4,2,4)
# drawCir(a,b,r,plot4)
pi4 = []
for i in range (100):
    pi4.append(calPai(n,plot4))
print("The value of Pi is : "+str(pi4)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(pi4))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(pi4))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(pi4))+" .\n")
```

```
# plot5
print("\n---------------------------------------------------\n")
n = 100
a,b,r = (0.,0.,1.)
# plt.figure(figsize=(15,15))
# plot5 = plt.subplot(4,2,5)
# drawCir(a,b,r,plot5)
pi5 = []
for i in range (100):
    pi5.append(calPai(n,plot5))
print("The value of Pi is : "+str(pi5)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(pi5))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(pi5))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(pi5))+" .\n")

# plot6
print("\n---------------------------------------------------\n")
n = 500
a,b,r = (0.,0.,1.)
# plt.figure(figsize=(15,15))
# plot6 = plt.subplot(4,2,6)
# drawCir(a,b,r,plot6)
pi6 = []
for i in range (100):
    pi6.append(calPai(n,plot6))
print("The value of Pi is : "+str(pi6)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(pi6))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(pi6))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(pi6))+" .\n")

# plot7
print("\n---------------------------------------------------\n")
n = 1000
a,b,r = (0.,0.,1.)
# plt.figure(figsize=(15,15))
# plot7 = plt.subplot(4,2,7)
# drawCir(a,b,r,plot7)
pi7 = []
for i in range (100):
    pi7.append(calPai(n,plot7))
print("The value of Pi is : "+str(pi7)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(pi7))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(pi7))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(pi7))+" .\n")

# plot8
print("\n---------------------------------------------------\n")
n = 5000
a,b,r = (0.,0.,1.)
# plt.figure(figsize=(15,15))
# plot8 = plt.subplot(4,2,8)
# drawCir(a,b,r,plot8)
pi8 = []
for i in range (100):
    pi8.append(calPai(n,plot8))
print("The value of Pi is : "+str(pi8)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(pi8))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(pi8))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(pi8))+" .\n")
```

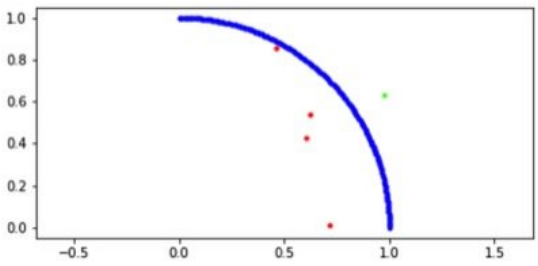
```
print("\n-----------------------------------------------------\n")
```

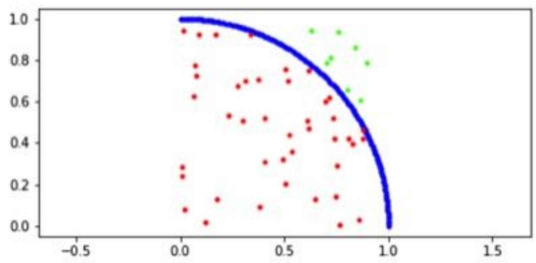
```
------------------------------------------------------

The value of Pi is : [3.2, 3.2, 1.6, 2.4, 4.0, 1.6, 3.2, 3.2, 3.2, 
4.0, 4.0, 2.4, 4.0, 3.2, 4.0, 0.8, 2.4, 3.2, 4.0, 4.0, 1.6, 3.2, 1.
6, 3.2, 3.2, 4.0, 2.4, 2.4, 2.4, 3.2, 4.0, 4.0, 2.4, 2.4, 4.0, 3.2, 
3.2, 2.4, 4.0, 4.0, 3.2, 3.2, 3.2, 3.2, 3.2, 2.4, 2.4, 2.4, 3.2, 3.
2, 3.2, 3.2, 2.4, 3.2, 3.2, 3.2, 4.0, 2.4, 1.6, 4.0, 4.0, 4.0, 3.2, 
2.4, 0.0, 4.0, 3.2, 4.0, 4.0, 4.0, 4.0, 3.2, 3.2, 3.2, 4.0, 1.6, 4.
0, 3.2, 4.0, 4.0, 4.0, 4.0, 3.2, 1.6, 4.0, 3.2, 3.2, 4.0, 2.4, 1.6, 
2.4, 4.0, 4.0, 3.2, 3.2, 3.2, 4.0, 2.4, 4.0, 3.2] .

Mean value of 5 points : 3.1439999999999992 .

The variance of 5 points : 0.6944639999999999 .

The standard variance of 5 points : 0.833345066584065 .


------------------------------------------------------

The value of Pi is : [2.8, 2.4, 3.6, 3.6, 3.2, 3.6, 2.4, 3.6, 3.2, 
2.8, 3.2, 3.6, 3.6, 2.8, 2.4, 3.2, 2.4, 1.6, 3.2, 3.6, 3.6, 2.8, 3.
2, 2.4, 2.8, 3.2, 2.8, 3.6, 3.2, 3.6, 3.2, 2.8, 3.6, 2.8, 3.6, 3.2, 
2.4, 3.6, 3.2, 3.6, 3.2, 3.6, 2.8, 3.2, 3.6, 3.6, 1.6, 3.6, 2.8, 2.
4, 3.2, 2.0, 3.6, 3.2, 3.2, 3.2, 2.8, 2.4, 2.8, 2.4, 2.0, 2.4, 2.8, 
3.2, 3.2, 3.6, 4.0, 3.6, 3.6, 3.2, 2.8, 2.4, 3.2, 3.2, 3.2, 3.6, 3.
2, 2.8, 2.8, 2.8, 3.6, 3.6, 3.2, 3.6, 2.4, 2.8, 2.8, 2.8, 3.2, 3.2, 
2.8, 3.2, 3.2, 2.8, 2.8, 3.2, 2.8, 2.8, 3.2, 3.6] .

Mean value of 10 points : 3.0640000000000005 .

The variance of 10 points : 0.22790400000000005 .

The standard variance of 10 points : 0.47739291993074223 .


------------------------------------------------------

The value of Pi is : [3.2, 3.2, 3.2, 3.4, 3.6, 2.8, 3.6, 3.4, 2.8, 
3.2, 2.8, 3.0, 3.2, 2.8, 2.6, 3.0, 3.6, 3.4, 3.6, 2.8, 3.8, 3.0, 3.
0, 3.0, 3.0, 3.8, 3.0, 3.2, 3.0, 2.6, 3.0, 2.6, 2.4, 2.6, 3.8, 2.8, 
3.2, 3.4, 3.4, 3.8, 3.6, 3.6, 3.2, 3.4, 3.0, 2.6, 3.2, 2.8, 2.8, 2.
8, 3.8, 3.2, 3.8, 3.2, 3.0, 3.4, 3.4, 2.4, 2.8, 3.2, 3.0, 3.6, 3.0, 
3.6, 3.2, 3.0, 3.4, 3.2, 3.4, 3.2, 3.6, 3.6, 3.2, 3.4, 3.0, 3.2, 3.
2, 3.6, 2.8, 3.4, 3.4, 3.0, 2.6, 3.4, 3.4, 2.8, 3.4, 3.2, 3.4, 2.4, 
3.4, 3.0, 2.8, 3.0, 3.2, 3.0, 3.4, 3.6, 2.8, 3.4] .

Mean value of 20 points : 3.17 .

The variance of 20 points : 0.11869999999999999 .

The standard variance of 20 points : 0.34452866353904427 .


------------------------------------------------------

The value of Pi is : [3.12, 3.44, 3.2, 3.2, 2.8, 3.52, 3.04, 3.12, 
3.36, 3.2, 3.44, 3.2, 2.96, 3.44, 3.44, 2.96, 2.96, 3.44, 3.12, 2.8
8, 2.72, 3.36, 3.2, 2.8, 2.96, 3.2, 2.88, 3.2, 3.2, 3.04, 2.96, 3.1
2, 3.44, 2.96, 3.2, 2.96, 2.96, 3.44, 3.12, 3.04, 3.36, 2.72, 2.96, 
3.12, 3.04, 3.2, 2.72, 3.36, 2.8, 3.12, 3.12, 2.88, 3.04, 3.04, 2.9
```

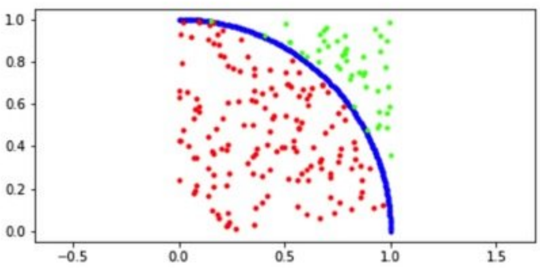
```
6, 3.36, 3.44, 3.04, 3.28, 3.12, 3.28, 3.36, 2.8, 3.44, 3.44, 2.88,
3.28, 2.96, 2.8, 2.8, 3.36, 2.72, 2.96, 2.88, 3.68, 3.2, 3.12, 2.88,
3.44, 3.28, 3.2, 3.28, 3.36, 2.72, 3.04, 3.12, 2.64, 3.28, 3.12, 3.
2, 3.52, 3.36, 3.12, 2.96, 3.12, 3.2, 3.28, 3.2, 3.04, 3.04] .

Mean value of 50 points : 3.1264 .

The variance of 50 points : 0.04936703999999999 .

The standard variance of 50 points : 0.22218694831155134 .


--------------------------------------------------------

The value of Pi is : [3.24, 2.76, 2.96, 3.2, 3.2, 3.2, 3.2, 3.28, 3.
2, 2.96, 2.72, 3.32, 3.32, 3.16, 3.08, 3.24, 2.96, 3.2, 3.12, 3.12,
3.28, 3.08, 3.0, 3.4, 3.08, 2.96, 3.48, 3.08, 2.96, 3.0, 3.44, 3.04,
3.36, 3.08, 3.12, 3.08, 3.28, 2.92, 2.96, 2.92, 2.92, 3.28, 3.08, 3.
28, 3.0, 3.12, 3.44, 2.92, 2.92, 3.2, 3.32, 3.32, 3.2, 2.88, 3.36,
3.24, 2.88, 3.0, 3.16, 3.16, 3.12, 3.32, 3.12, 3.4, 3.32, 3.08, 3.2,
3.24, 2.96, 3.04, 3.16, 3.04, 3.4, 3.24, 2.72, 3.52, 3.16, 3.12, 3.2
8, 2.96, 3.2, 3.16, 3.08, 2.88, 3.32, 3.16, 2.72, 2.96, 2.84, 3.08,
2.92, 3.04, 3.0, 2.96, 3.0, 3.16, 3.04, 3.36, 3.36, 3.2] .

Mean value of 100 points : 3.1248 .

The variance of 100 points : 0.030376959999999998 .

The standard variance of 100 points : 0.1742898734866716 .


--------------------------------------------------------

The value of Pi is : [3.064, 3.216, 3.256, 3.088, 3.064, 3.176, 3.17
6, 3.176, 3.064, 3.04, 3.056, 2.992, 2.992, 3.152, 3.096, 3.232, 3.1
36, 3.16, 3.304, 3.016, 3.104, 3.232, 3.256, 3.184, 3.248, 3.192, 3.
208, 3.208, 3.08, 3.2, 3.208, 3.208, 3.176, 3.144, 3.112, 3.168, 3.1
6, 3.208, 3.096, 3.024, 3.144, 3.128, 3.208, 3.168, 3.216, 3.288, 3.
256, 3.096, 3.104, 3.16, 3.064, 3.152, 3.128, 3.104, 3.064, 3.176,
3.144, 3.216, 3.152, 3.224, 3.296, 3.16, 3.312, 3.136, 3.176, 3.176,
3.16, 3.136, 3.056, 3.152, 3.016, 3.088, 3.048, 3.16, 3.248, 3.208,
3.192, 3.104, 3.08, 3.088, 3.272, 3.168, 3.232, 3.088, 3.248, 3.112,
3.136, 3.224, 3.296, 3.16, 3.312, 3.144, 3.192, 3.176, 3.112, 3.264,
3.04, 3.16, 2.984, 3.048] .

Mean value of 500 points : 3.15424 .

The variance of 500 points : 0.005879142399999997 .

The standard variance of 500 points : 0.07667556586031822 .


--------------------------------------------------------

The value of Pi is : [3.252, 3.168, 3.2, 3.144, 3.156, 3.216, 3.148,
3.08, 3.084, 3.144, 3.172, 3.068, 3.1, 3.184, 3.156, 3.108, 3.156,
3.028, 3.2, 3.096, 3.068, 3.088, 3.132, 3.132, 3.18, 3.12, 3.112, 3.
144, 3.136, 3.124, 3.124, 3.096, 3.124, 3.144, 3.108, 3.224, 3.164,
3.256, 3.1, 3.208, 3.1, 3.152, 3.12, 3.132, 3.188, 3.172, 3.188, 3.1
52, 3.172, 3.2, 3.224, 3.108, 3.204, 3.196, 3.14, 3.152, 3.108, 3.21
2, 3.232, 3.108, 3.092, 3.232, 3.116, 3.112, 3.184, 3.152, 3.132, 3.
```

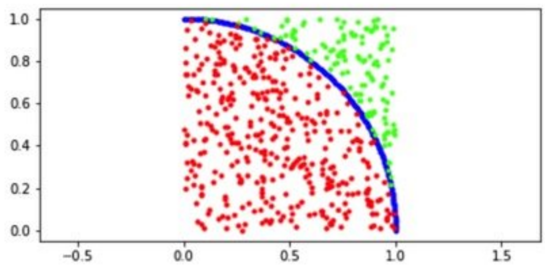
```
116, 3.176, 3.18, 3.148, 3.176, 3.18, 3.172, 3.144, 3.18, 3.128, 3.1
44, 3.184, 3.088, 3.124, 3.148, 3.232, 3.208, 3.068, 3.164, 3.108,
3.076, 3.184, 3.1, 3.124, 3.116, 3.096, 3.228, 3.152, 3.168, 3.084,
3.24, 3.144, 3.16] .

Mean value of 1000 points : 3.1486400000000003 .

The variance of 1000 points : 0.002182950400000002 .

The standard variance of 1000 points : 0.04672205474933655 .


-------------------------------------------------------

The value of Pi is : [3.1352, 3.1144, 3.1656, 3.1832, 3.1488, 3.144
8, 3.1432, 3.1632, 3.152, 3.1016, 3.1752, 3.1448, 3.1192, 3.136, 3.1
824, 3.176, 3.0976, 3.1672, 3.156, 3.1608, 3.1632, 3.1608, 3.0864,
3.136, 3.1176, 3.1752, 3.1128, 3.1736, 3.1544, 3.1576, 3.1736, 3.109
6, 3.1592, 3.148, 3.1264, 3.1144, 3.1672, 3.1384, 3.1312, 3.1576, 3.
1096, 3.128, 3.1392, 3.1352, 3.1072, 3.1368, 3.1832, 3.152, 3.1336,
3.1256, 3.1912, 3.1608, 3.1368, 3.1272, 3.1424, 3.1288, 3.1008, 3.15
68, 3.1472, 3.1312, 3.1112, 3.152, 3.132, 3.132, 3.1608, 3.1864, 3.1
112, 3.1664, 3.1344, 3.1472, 3.1136, 3.1112, 3.1168, 3.1112, 3.172,
3.1216, 3.1184, 3.1672, 3.1008, 3.1064, 3.1504, 3.1656, 3.1712, 3.14
16, 3.1496, 3.1656, 3.16, 3.1416, 3.1328, 3.1496, 3.1232, 3.0912, 3.
1296, 3.1472, 3.1632, 3.172, 3.1664, 3.132, 3.1248, 3.1872] .

Mean value of 5000 points : 3.1424160000000008 .

The variance of 5000 points : 0.0005916541439999983 .

The standard variance of 5000 points : 0.02432394178582078 .


-------------------------------------------------------
```

Excerpts of results are shown below.









Experiment results are shown and enhanced experiment results are stored in **data1(_1).xlsx.**

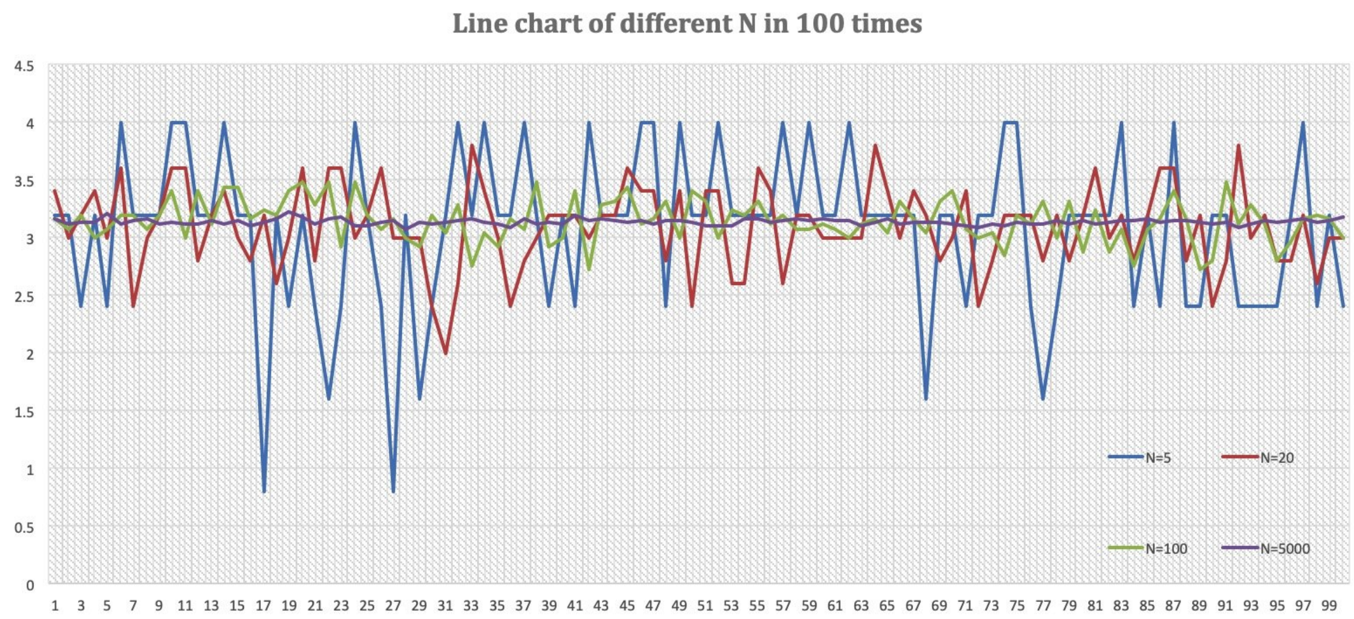
Further Analysis

In [91]:

```
df1 = pd.DataFrame({'Iteration':range(100),'N=5':pi1,'N=20':pi3,'N=100':pi5,'N=5
000':pi8})
df1.to_excel('data1.xlsx')

N = [5,10,20,50,100,500,1000,5000]

mean = []
mean.append(np.mean(pi1))
mean.append(np.mean(pi2))
mean.append(np.mean(pi3))
mean.append(np.mean(pi4))
mean.append(np.mean(pi5))
mean.append(np.mean(pi6))
mean.append(np.mean(pi7))
mean.append(np.mean(pi8))

variance = []
variance.append(np.var(pi1))
variance.append(np.var(pi2))
variance.append(np.var(pi3))
variance.append(np.var(pi4))
variance.append(np.var(pi5))
variance.append(np.var(pi6))
variance.append(np.var(pi7))
variance.append(np.var(pi8))

standard = []
standard.append(np.std(pi1))
standard.append(np.std(pi2))
standard.append(np.std(pi3))
standard.append(np.std(pi4))
standard.append(np.std(pi5))
standard.append(np.std(pi6))
standard.append(np.std(pi7))
standard.append(np.std(pi8))

df1_1 = pd.DataFrame({'Points':N,'Mean':mean,'Variance':variance,'Standard':stan
dard})
df1_1.to_excel('data1_1.xlsx')
```

To systematically analyze our experiment results, a great amount of data is of necessity. So we scaled up the experiment and even go so far as to 5000 points in 100 times. Enhanced experiment results are shown in the figure below. The huge amount of data illuminates us some important ideas:

- First, it can be seen that the system variance of calculation become smaller when we conduct our experiment with more random points.
- With the cement data support, we can get the value of Pi stabilizes around 3.1~3.2, which is concord with our common recognition.



Points	5	10	20	50	100	500	1000	5000
Mean	3.144	3.064	3.17	3.1264	3.1248	3.15424	3.14864	3.142416
Variance	0.694464	0.227904	0.1187	0.04936704	0.03037696	0.005879142	0.00218295	0.000591654
Standard	0.833345067	0.47739292	0.344528664	0.222186948	0.174289873	0.076675566	0.046722055	0.024323942

Excercise 2

Question Description

We are now trying to integrate another function by the Monte Carlo method:

$$\int x^3$$

A simple analytic solution exists here:

$$\int_0^1 x^3 = \frac{1}{4}$$

If you compute this integration using the Monte Carlo method, what distribution do you use to sample x? How good do you get when N = 5, 10, 20, 30, 40, 50, 60, 70, 80, 100, respectively? For each N, repeat the Monte Carlo process 20 times, and report the mean and variance of the integrate in a table.

Implementation Details

- In the first part, trying to have a clear overview of the whole method, we use [matplotlib (https://www.baidu.com/link?url=AkPiPYbYiYO0fQwZi0pVbV7UvxFba4Bn6uHBHPOWsim&wd=&eqid=d5c1106300015827000000065c](https://www.baidu.com/link?url=AkPiPYbYiYO0fQwZi0pVbV7UvxFba4Bn6uHBHPOWsim&wd=&eqid=d5c1106300015827000000065c) to draw the random distribution of N observations in the given area. Images are slowly shown, so we have to try a limit amount of samples in this part.
- In order to further explore this state of method and give the analysis of the experiment results, we increase the times of repetitive experiments and use a huge amount of observations. Luckily, the experiment is well-conduct and show results as we expected. N = [5,10,20,50,100,500,1000,5000] correspondingly.

Monte Carlo

In [64]:

```
import random
import numpy as np
import matplotlib.pyplot as plt

def drawFunc(plot_num):
    x = np.linspace(0,1,500)
    y = np.power(x,3)
#     plot_num.plot([x],[y],'b.')
#     plot_num.axis('equal')


def calX3(n,plot_num=None):
    r = 1.0
    a,b = (0.0,0.0)
    x_pos = a+r
    y_pos = b+r

    count = 0
    for i in range(0,n):
        x = random.uniform(0,x_pos)
        y = random.uniform(0,y_pos)
        if x*x*x >= y:
            count += 1
#             plot_num.plot([x],[y],'r.')
            pass
        else:
#             plot_num.plot([x],[y],color='#40fd14',marker='.')
            pass

    val = (count/float(n))
    plt.show()
    return val
```

Experiment Results

In [102]:

```
# plot1
print("\n--------------------------------------------------------\n")
n = 5
# plt.figure(figsize=(15,15))
# plot1 = plt.subplot(5,2,1)
# drawFunc(plot1)
res1 = []
for i in range (100):
    res1.append(calX3(n,plot1))
print("The integration of X3 from 0 to 1 is : "+str(res1)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res1))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res1))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res1))+" .\n")

# plot2
print("\n--------------------------------------------------------\n")
n = 10
# plt.figure(figsize=(15,15))
# plot2 = plt.subplot(5,2,2)
# drawFunc(plot2)
res2 = []
for i in range (100):
    res2.append(calX3(n,plot2))
print("The integration of X3 from 0 to 1 is : "+str(res2)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res2))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res2))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res2))+" .\n")

# plot3
print("\n--------------------------------------------------------\n")
n = 20
# plt.figure(figsize=(15,15))
# plot3 = plt.subplot(5,2,3)
# drawFunc(plot3)
res3 = []
for i in range (100):
    res3.append(calX3(n,plot3))
print("The integration of X3 from 0 to 1 is : "+str(res3)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res3))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res3))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res3))+" .\n")

# plot4
print("\n--------------------------------------------------------\n")
n = 50
# plt.figure(figsize=(15,15))
# plot4 = plt.subplot(5,2,4)
# drawFunc(plot4)
res4 = []
for i in range (100):
    res4.append(calX3(n,plot4))
print("The integration of X3 from 0 to 1 is : "+str(res4)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res4))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res4))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res4))+" .\n")

# plot5
print("\n--------------------------------------------------------\n")
n = 100
```

```
# plt.figure(figsize=(15,15))
# plot5 = plt.subplot(5,2,5)
# drawFunc(plot5)
res5 = []
for i in range (100):
    res5.append(calX3(n,plot5))
print("The integration of X3 from 0 to 1 is : "+str(res5)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res5))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res5))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res5))+" .\n")

# plot6
print("\n-------------------------------------------------------\n")
n = 500
# plt.figure(figsize=(15,15))
# plot6 = plt.subplot(5,2,6)
# drawFunc(plot6)
res6 = []
for i in range (100):
    res6.append(calX3(n,plot6))
print("The integration of X3 from 0 to 1 is : "+str(res6)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res6))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res6))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res6))+" .\n")

# plot7
print("\n-------------------------------------------------------\n")
n = 1000
# plt.figure(figsize=(15,15))
# plot7 = plt.subplot(5,2,7)
# drawFunc(plot7)
res7 = []
for i in range (100):
    res7.append(calX3(n,plot7))
print("The integration of X3 from 0 to 1 is : "+str(res7)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res7))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res7))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res7))+" .\n")

# plot8
print("\n-------------------------------------------------------\n")
n = 5000
# plt.figure(figsize=(15,15))
# plot8 = plt.subplot(5,2,8)
# drawFunc(plot8)
res8 = []
for i in range (100):
    res8.append(calX3(n,plot8))
print("The integration of X3 from 0 to 1 is : "+str(res8)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res8))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res8))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res8))+" .\n")

# plot9
print("\n-------------------------------------------------------\n")
n = 10000
# plt.figure(figsize=(15,15))
# plot9 = plt.subplot(5,2,9)
# drawFunc(plot9)
res9 = []
for i in range (100):
```

```
    res9.append(calX3(n))
print("The integration of X3 from 0 to 1 is : "+str(res9)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res9))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res9))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res9))+" .\n")

# plot10
print("\n-------------------------------------------------------\n")
n = 100000
# plt.figure(figsize=(15,15))
# plot10 = plt.subplot(5,2,10)
# drawFunc(plot10)
res10 = []
for i in range (100):
    res10.append(calX3(n))
print("The integration of X3 from 0 to 1 is : "+str(res10)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res10))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res10))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res10))+" .\n")

print("\n-------------------------------------------------------\n")
```

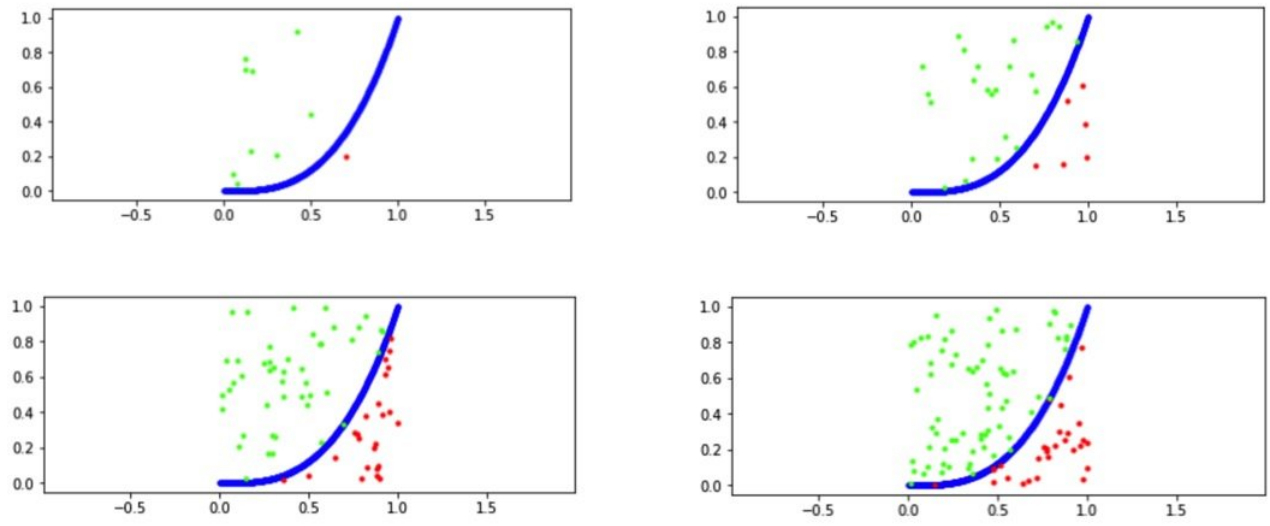
```
------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.0, 0.2, 0.4, 0.2, 0.0, 0.
0, 0.2, 0.2, 0.0, 0.2, 0.4, 0.4, 0.2, 0.2, 0.4, 0.2, 0.4, 0.2, 0.0,
0.4, 0.2, 0.2, 0.4, 0.2, 0.4, 0.2, 0.0, 0.6, 0.6, 0.6, 0.2, 0.4, 0.
4, 0.6, 0.4, 0.4, 0.2, 0.2, 0.2, 0.0, 0.4, 0.4, 0.0, 0.0, 0.6, 0.2,
0.2, 0.2, 0.2, 0.2, 0.2, 0.0, 0.2, 0.0, 0.4, 0.2, 0.2, 0.4, 0.2, 0.
0, 0.4, 0.0, 0.6, 0.0, 0.2, 0.0, 0.2, 0.2, 0.6, 0.4, 0.8, 0.0, 0.4,
0.4, 0.4, 0.2, 0.2, 0.2, 0.0, 0.2, 0.6, 0.2, 0.0, 0.0, 0.0, 0.2, 0.
4, 0.2, 0.4, 0.6, 0.4, 0.6, 0.2, 0.4, 0.2, 0.2, 0.2, 0.2, 0.2, 0.0]
.

Mean value of 5 points : 0.254 .

The variance of 5 points : 0.035084 .

The standard variance of 5 points : 0.18730723424363513 .


------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.1, 0.1, 0.1, 0.2, 0.2, 0.
2, 0.3, 0.3, 0.2, 0.2, 0.1, 0.5, 0.2, 0.5, 0.2, 0.2, 0.3, 0.3, 0.2,
0.3, 0.3, 0.2, 0.7, 0.2, 0.2, 0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.3, 0.
2, 0.2, 0.5, 0.3, 0.0, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.6, 0.3, 0.2,
0.2, 0.3, 0.3, 0.5, 0.3, 0.1, 0.2, 0.6, 0.3, 0.4, 0.0, 0.1, 0.4, 0.
2, 0.2, 0.1, 0.4, 0.2, 0.3, 0.4, 0.2, 0.5, 0.0, 0.3, 0.2, 0.4, 0.3,
0.1, 0.1, 0.3, 0.3, 0.4, 0.4, 0.4, 0.2, 0.3, 0.0, 0.4, 0.4, 0.2, 0.
0, 0.3, 0.2, 0.2, 0.2, 0.3, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.0, 0.5]
.

Mean value of 10 points : 0.254 .

The variance of 10 points : 0.018483999999999997 .

The standard variance of 10 points : 0.13595587519485872 .


------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.35, 0.25, 0.05, 0.15, 0.1,
0.25, 0.2, 0.2, 0.35, 0.2, 0.1, 0.2, 0.25, 0.15, 0.45, 0.15, 0.2, 0.
2, 0.2, 0.1, 0.25, 0.2, 0.05, 0.45, 0.25, 0.3, 0.4, 0.2, 0.45, 0.25,
0.2, 0.15, 0.15, 0.35, 0.35, 0.2, 0.35, 0.35, 0.1, 0.35, 0.15, 0.35,
0.4, 0.2, 0.2, 0.25, 0.35, 0.4, 0.25, 0.35, 0.2, 0.3, 0.0, 0.15, 0.
1, 0.4, 0.1, 0.25, 0.3, 0.35, 0.25, 0.2, 0.15, 0.1, 0.2, 0.3, 0.4,
0.2, 0.15, 0.15, 0.35, 0.1, 0.15, 0.2, 0.2, 0.35, 0.35, 0.35, 0.15,
0.3, 0.35, 0.3, 0.25, 0.1, 0.4, 0.2, 0.2, 0.15, 0.2, 0.35, 0.45, 0.
1, 0.3, 0.3, 0.25, 0.1, 0.3, 0.45, 0.05, 0.25] .

Mean value of 20 points : 0.24150000000000002 .

The variance of 20 points : 0.011352750000000002 .

The standard variance of 20 points : 0.10654928437113034 .


------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.18, 0.16, 0.34, 0.12, 0.2
4, 0.3, 0.14, 0.36, 0.28, 0.32, 0.24, 0.24, 0.24, 0.26, 0.28, 0.32,
```

```
0.26, 0.28, 0.26, 0.26, 0.22, 0.34, 0.26, 0.26, 0.32, 0.32, 0.28, 0.
2, 0.24, 0.2, 0.26, 0.18, 0.22, 0.2, 0.2, 0.24, 0.32, 0.22, 0.2, 0.3
4, 0.24, 0.16, 0.34, 0.36, 0.32, 0.28, 0.34, 0.24, 0.26, 0.2, 0.36,
0.28, 0.24, 0.24, 0.26, 0.3, 0.26, 0.32, 0.16, 0.18, 0.18, 0.4, 0.3,
0.18, 0.2, 0.16, 0.18, 0.24, 0.36, 0.24, 0.14, 0.26, 0.14, 0.28, 0.2
8, 0.28, 0.18, 0.32, 0.24, 0.3, 0.26, 0.24, 0.18, 0.22, 0.14, 0.24,
0.26, 0.26, 0.22, 0.18, 0.2, 0.34, 0.26, 0.34, 0.12, 0.28, 0.16, 0.2
8, 0.32, 0.34] .

Mean value of 50 points : 0.25140000000000007 .

The variance of 50 points : 0.00393804 .

The standard variance of 50 points : 0.06275380466553403 .


-------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.22, 0.24, 0.26, 0.25, 0.1
5, 0.26, 0.25, 0.29, 0.3, 0.26, 0.25, 0.21, 0.28, 0.23, 0.24, 0.21,
0.17, 0.24, 0.26, 0.24, 0.27, 0.23, 0.22, 0.25, 0.16, 0.23, 0.18, 0.
24, 0.21, 0.29, 0.27, 0.37, 0.21, 0.2, 0.32, 0.31, 0.25, 0.16, 0.2,
0.3, 0.27, 0.16, 0.23, 0.25, 0.3, 0.26, 0.24, 0.31, 0.29, 0.28, 0.2
1, 0.26, 0.21, 0.21, 0.31, 0.2, 0.26, 0.22, 0.23, 0.16, 0.31, 0.23,
0.25, 0.28, 0.3, 0.22, 0.26, 0.25, 0.23, 0.24, 0.26, 0.25, 0.32, 0.2
6, 0.22, 0.35, 0.25, 0.15, 0.28, 0.25, 0.23, 0.22, 0.22, 0.23, 0.24,
0.25, 0.27, 0.33, 0.26, 0.28, 0.28, 0.24, 0.27, 0.26, 0.27, 0.22, 0.
25, 0.27, 0.22, 0.25] .

Mean value of 100 points : 0.24760000000000001 .

The variance of 100 points : 0.0017362399999999998 .

The standard variance of 100 points : 0.04166821330462826 .


-------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.258, 0.266, 0.24, 0.282,
0.234, 0.232, 0.236, 0.248, 0.266, 0.21, 0.268, 0.226, 0.244, 0.226,
0.23, 0.25, 0.25, 0.246, 0.242, 0.22, 0.238, 0.258, 0.25, 0.25, 0.22
4, 0.24, 0.222, 0.258, 0.288, 0.264, 0.276, 0.246, 0.27, 0.246, 0.22
6, 0.29, 0.224, 0.26, 0.238, 0.24, 0.256, 0.254, 0.27, 0.234, 0.24,
0.236, 0.3, 0.278, 0.28, 0.248, 0.288, 0.254, 0.284, 0.222, 0.238,
0.27, 0.278, 0.266, 0.25, 0.242, 0.254, 0.25, 0.266, 0.232, 0.258,
0.218, 0.254, 0.24, 0.248, 0.248, 0.256, 0.26, 0.244, 0.256, 0.28,
0.278, 0.276, 0.25, 0.216, 0.238, 0.294, 0.238, 0.222, 0.274, 0.256,
0.248, 0.28, 0.258, 0.268, 0.23, 0.282, 0.274, 0.262, 0.284, 0.218,
0.27, 0.266, 0.268, 0.234, 0.278] .

Mean value of 500 points : 0.25296 .

The variance of 500 points : 0.00040899840000000007 .

The standard variance of 500 points : 0.020223708858663883 .


-------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.251, 0.224, 0.269, 0.244,
0.261, 0.263, 0.26, 0.226, 0.262, 0.235, 0.254, 0.239, 0.235, 0.257,
```

```
0.239, 0.254, 0.242, 0.221, 0.276, 0.253, 0.241, 0.222, 0.235, 0.24
2, 0.234, 0.258, 0.26, 0.236, 0.258, 0.261, 0.259, 0.26, 0.247, 0.25
5, 0.257, 0.258, 0.253, 0.231, 0.242, 0.24, 0.224, 0.223, 0.258, 0.2
48, 0.264, 0.246, 0.25, 0.253, 0.268, 0.229, 0.263, 0.244, 0.278, 0.
231, 0.255, 0.268, 0.229, 0.243, 0.224, 0.263, 0.257, 0.251, 0.242,
0.255, 0.264, 0.263, 0.285, 0.259, 0.248, 0.254, 0.232, 0.235, 0.25
7, 0.28, 0.25, 0.25, 0.251, 0.242, 0.259, 0.249, 0.242, 0.229, 0.24
9, 0.277, 0.229, 0.233, 0.232, 0.249, 0.222, 0.235, 0.226, 0.229, 0.
257, 0.269, 0.275, 0.235, 0.249, 0.263, 0.269, 0.229] .

Mean value of 1000 points : 0.24856 .

The variance of 1000 points : 0.00022672640000000012 .

The standard variance of 1000 points : 0.015057436700846532 .


-------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.2574, 0.2486, 0.251, 0.252
8, 0.2506, 0.2496, 0.2496, 0.2522, 0.2552, 0.2446, 0.2474, 0.2546,
0.2526, 0.2488, 0.2528, 0.246, 0.248, 0.2376, 0.2552, 0.2518, 0.244
4, 0.2402, 0.2438, 0.2484, 0.2494, 0.253, 0.2508, 0.2444, 0.2432, 0.
2406, 0.2588, 0.25, 0.2468, 0.2498, 0.2488, 0.245, 0.2388, 0.2418,
0.2446, 0.2394, 0.2418, 0.2464, 0.2504, 0.2508, 0.2564, 0.2398, 0.25
36, 0.2562, 0.2568, 0.2612, 0.2512, 0.2436, 0.2412, 0.2576, 0.2576,
0.2542, 0.2462, 0.2504, 0.2374, 0.2488, 0.2386, 0.2548, 0.2408, 0.25
08, 0.2512, 0.2544, 0.2506, 0.2436, 0.2464, 0.2548, 0.2494, 0.259,
0.2474, 0.2544, 0.2422, 0.269, 0.2474, 0.249, 0.2422, 0.2568, 0.253
6, 0.2564, 0.2524, 0.2324, 0.2484, 0.2548, 0.2626, 0.2438, 0.252, 0.
249, 0.2408, 0.2436, 0.2422, 0.2424, 0.25, 0.2486, 0.2448, 0.2564,
0.2548, 0.2502] .

Mean value of 5000 points : 0.24911999999999998 .

The variance of 5000 points : 3.88248e-05 .

The standard variance of 5000 points : 0.006230954982986155 .


-------------------------------------------------------

The integration of X3 from 0 to 1 is : [0.2513, 0.2519, 0.2409, 0.25
27, 0.2542, 0.2519, 0.2508, 0.2446, 0.2504, 0.2456, 0.2547, 0.2532,
0.2526, 0.2513, 0.24, 0.2487, 0.2505, 0.2526, 0.2553, 0.2534, 0.253
3, 0.2535, 0.2502, 0.2488, 0.2396, 0.2522, 0.2517, 0.2493, 0.2531,
0.2525, 0.2497, 0.2542, 0.253, 0.2505, 0.2527, 0.2473, 0.2499, 0.239
4, 0.2426, 0.2465, 0.2525, 0.2496, 0.2408, 0.2445, 0.2504, 0.2545,
0.2548, 0.2548, 0.2507, 0.2556, 0.2411, 0.2487, 0.2399, 0.253, 0.245
8, 0.2441, 0.2495, 0.244, 0.2548, 0.251, 0.2504, 0.2504, 0.2477, 0.2
482, 0.2491, 0.2496, 0.2482, 0.2469, 0.2498, 0.2517, 0.2425, 0.249,
0.2489, 0.2544, 0.2495, 0.2509, 0.2447, 0.2479, 0.2531, 0.2519, 0.24
19, 0.2554, 0.2528, 0.2485, 0.2496, 0.245, 0.2535, 0.2517, 0.2446,
0.2472, 0.2434, 0.2497, 0.2496, 0.2461, 0.252, 0.2556, 0.2512, 0.24
1, 0.2485, 0.2492] .

Mean value of 10000 points : 0.24938 .

The variance of 10000 points : 1.7308200000000005e-05 .

The standard variance of 10000 points : 0.004160312488263352 .
```

```
----------------------------------------------------

The integration of X3 from 0 to 1 is : [0.24981, 0.24979, 0.24951,
0.25116, 0.24972, 0.25139, 0.25297, 0.24865, 0.24935, 0.25102, 0.248
94, 0.25038, 0.2498, 0.25102, 0.25134, 0.24873, 0.24872, 0.24822, 0.
24829, 0.25113, 0.25047, 0.24816, 0.25005, 0.25115, 0.24909, 0.2509
2, 0.2509, 0.25088, 0.24896, 0.24824, 0.25096, 0.25108, 0.24881, 0.2
4954, 0.25106, 0.25089, 0.25241, 0.25005, 0.24734, 0.24975, 0.24891,
0.24913, 0.24693, 0.25057, 0.25069, 0.24891, 0.2508, 0.24987, 0.2497
4, 0.24817, 0.25027, 0.25138, 0.25011, 0.25099, 0.24795, 0.24927, 0.
25044, 0.25153, 0.2505, 0.24885, 0.25143, 0.2494, 0.24989, 0.25182,
0.24978, 0.24954, 0.25086, 0.24991, 0.25097, 0.25056, 0.25053, 0.250
16, 0.2526, 0.25261, 0.25091, 0.24963, 0.25189, 0.24814, 0.24876, 0.
2492, 0.24984, 0.24955, 0.24842, 0.25181, 0.25013, 0.24902, 0.2489,
0.24642, 0.25063, 0.25052, 0.25028, 0.24968, 0.25115, 0.25039, 0.250
48, 0.24779, 0.24815, 0.25154, 0.24994, 0.24842] .

Mean value of 100000 points : 0.24997309999999995 .

The variance of 100000 points : 1.5944913899999986e-06 .

The standard variance of 100000 points : 0.0012627317173493334 .

----------------------------------------------------
```

Excerpts of results are shown below.



Experiment results are shown and enhanced experiment results are stored in **data2(_1).xlsx.**

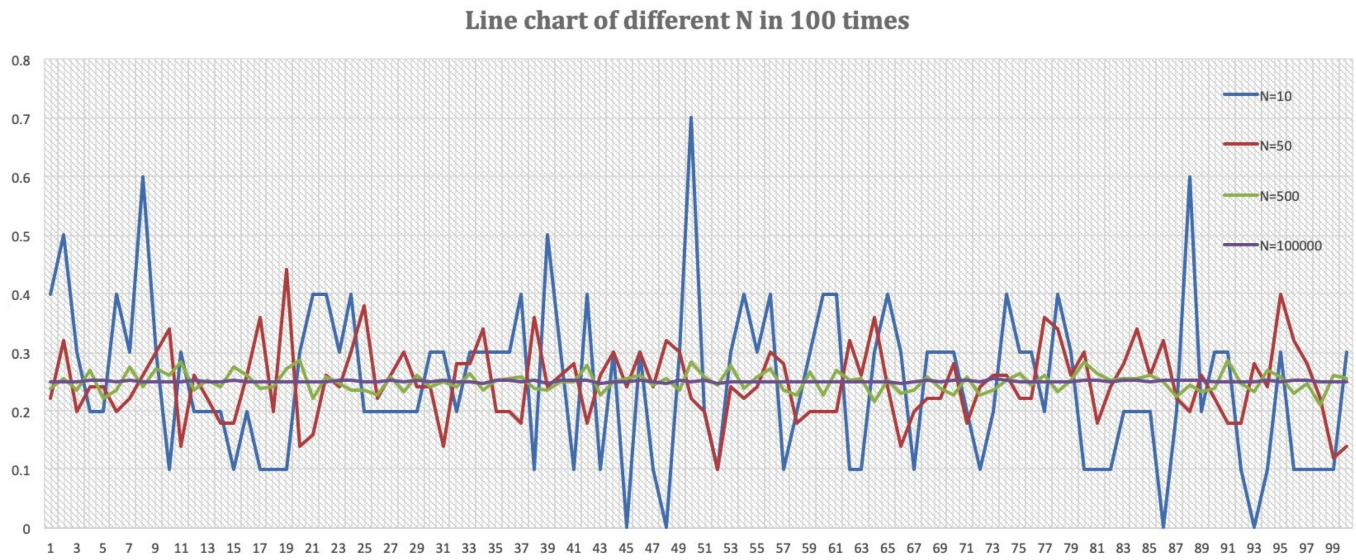
Further Analysis

In [104]:

```
df2 = pd.DataFrame({'Iteration':range(100),'N=10':res2,'N=50':res4,'N=500':res6,
'N=100000':res10})
df2.to_excel('data2.xlsx')

N = [5,10,20,50,100,500,1000,5000,10000,100000]

mean = []
mean.append(np.mean(res1))
mean.append(np.mean(res2))
mean.append(np.mean(res3))
mean.append(np.mean(res4))
mean.append(np.mean(res5))
mean.append(np.mean(res6))
mean.append(np.mean(res7))
mean.append(np.mean(res8))
mean.append(np.mean(res9))
mean.append(np.mean(res10))

variance = []
variance.append(np.var(res1))
variance.append(np.var(res2))
variance.append(np.var(res3))
variance.append(np.var(res4))
variance.append(np.var(res5))
variance.append(np.var(res6))
variance.append(np.var(res7))
variance.append(np.var(res8))
variance.append(np.var(res9))
variance.append(np.var(res10))

standard = []
standard.append(np.std(res1))
standard.append(np.std(res2))
standard.append(np.std(res3))
standard.append(np.std(res4))
standard.append(np.std(res5))
standard.append(np.std(res6))
standard.append(np.std(res7))
standard.append(np.std(res8))
standard.append(np.std(res9))
standard.append(np.std(res10))

df2 = pd.DataFrame({'Points':N,'Mean':mean,'Variance':variance,'Standard':standa
rd})
df2.to_excel('data2_1.xlsx')
```

To systematically analyze our experiment results, a great amount of data is of necessity. So we scaled up the experiment and even go so far as to 100000 points in 100 times. Enhanced experiment results are shown in the figure below. The huge amount of data illuminates us some important ideas:

- First, it can be seen that the system variance of calculation become smaller when we conduct our experiment with more random points.
- With the cement data support, we can get the value of result stablizes around 2.5, which is concord with our common recognition.



Points	5	10	20	50	100	500	1000	5000	10000	100000
Mean	0.254	0.254	0.2415	0.2514	0.2476	0.25296	0.24856	0.24912	0.24938	0.2499731
Variance	0.035084	0.018484	0.01135275	0.00393804	0.00173624	0.000408998	0.000226726	3.88248E-05	1.73082E-05	1.59449E-06
Standard	0.187307234	0.135955875	0.106549284	0.062753805	0.041668213	0.020223709	0.015057437	0.006230955	0.004160312	0.001262732

Excercise 3

Question Description

We are now trying to integrate a more difficult function by the Monte Carlo method that may not be analytically computed:

$$\int_{x=2}^{4}\int_{y=-1}^{1} f(x,y) = \frac{y^2 * e^{-y^2} x^4 * e^{-x^2}}{x * e^{-x^2}}$$

Can you compute the above integration analytically? If you compute this integration using the Monte Carlo method, what distribution do you use to sample (x,y)? How good do you get when the sample sizes are N = 5, 10, 20, 30, 40, 50, 60, 70, 80, 100, 200 respectively? For each N, repeat the Monte Carlo process 100 times, and report the mean and variance of the integration.

Implementation Details

- In the first part, trying to have a clear overview of the whole method, we use matplotlib to draw the random distribution of N observations in the given area. Images are slowly shown, so we have to try a limit amount of samples in this part.
- In order to further explore this state of method and give the analysis of the experiment results, we increase the times of repetitive experiments and use a huge amount of observations. Luckily, the experiment is well-conduct and show results as we expected. N = [5,10,20,50,100,500,1000,5000] correspondingly.
- This exercise is a double integration problem. As a result, the calculation is more sophisticated. We obtain the properties of this integrated function by picturing its 3-D contour and draw the conclusion that we get the max value when x = 4, y = +/-1, where f(4,+/-1) = 817318.34.

Monte Carlo

In [71]:

```
import random
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from matplotlib import cm

def drawFunc(ax_num):
    x,y = np.meshgrid(np.linspace(2,4,10),np.linspace(-1,1,10))
    z = (y*y*np.exp(-y*y)+np.power(x,4)*np.exp(-x*x))/(x*np.exp(-x*x))
#     ax_num.contourf(x,y,z,100,cmap=plt.get_cmap('Blues'))


def calIntInt(n,plot_num):

    count = 0
    for i in range(0,n):
        x = random.uniform(2,4)
        y = random.uniform(-1,1)
        max = 817318.3431180277
        z = random.uniform(0,817318.3431180277)
        if (y*y*np.exp(-y*y)+np.power(x,4)*np.exp(-x*x))/(x*np.exp(-x*x)) >= z:
            count += 1
#             ax.scatter(x,y,z,marker='.',color='r')
            pass
        else:
#             ax.scatter(x,y,z,marker='.',color='#40fd14')
            pass

    val = (count/float(n))*4*max
    plt.show()
    return val
```

Experiment Results

In [106]:

```
# plot1
print("\n------------------------------------------------------\n")
n = 5
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
# drawFunc(ax)
res1 = []
for i in range (50):
    res1.append(calIntInt(n,ax))
print("The double integration is : "+str(res1)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res1))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res1))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res1))+" .\n")

# plot2
print("\n------------------------------------------------------\n")
n = 10
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
# drawFunc(ax)
res2 = []
for i in range (50):
    res2.append(calIntInt(n,ax))
print("The double integration is : "+str(res2)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res2))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res2))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res2))+" .\n")

# plot3
print("\n------------------------------------------------------\n")
n = 20
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
# drawFunc(ax)
res3 = []
for i in range (50):
    res3.append(calIntInt(n,ax))
print("The double integration is : "+str(res3)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res3))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res3))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res3))+" .\n")

# plot4
print("\n------------------------------------------------------\n")
n = 50
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
# drawFunc(ax)
res4 = []
for i in range (50):
    res4.append(calIntInt(n,ax))
```

```
print("The double integration is : "+str(res4)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res4))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res4))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res4))+" .\n")

# plot5
print("\n-------------------------------------------------------\n")
n = 100
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
# drawFunc(ax)
res5 = []
for i in range (50):
    res5.append(calIntInt(n,ax))
print("The double integration is : "+str(res5)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res5))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res5))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res5))+" .\n")

# plot6
print("\n-------------------------------------------------------\n")
n = 500
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
#drawFunc(ax)
res6 = []
for i in range (50):
    res6.append(calIntInt(n,ax))
print("The double integration is : "+str(res6)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res6))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res6))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res6))+" .\n")

# plot7
print("\n-------------------------------------------------------\n")
n = 1000
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
# drawFunc(ax)
res7 = []
for i in range (50):
    res7.append(calIntInt(n,ax))
print("The double integration is : "+str(res7)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res7))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res7))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res7))+" .\n")

# plot8
print("\n-------------------------------------------------------\n")
n = 5000
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
#drawFunc(ax)
```

```
res8 = []
for i in range (50):
    res8.append(calIntInt(n,ax))
print("The double integration is : "+str(res8)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res8))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res8))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res8))+" .\n")

# plot9
print("\n-------------------------------------------------------\n")
n = 10000
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
#drawFunc(ax)
res9 = []
for i in range (50):
    res9.append(calIntInt(n,ax))
print("The double integration is : "+str(res9)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res9))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res9))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res9))+" .\n")

# plot10
print("\n-------------------------------------------------------\n")
n = 100000
# ax = plt.gca(projection='3d')
# ax.set_xlabel('x', fontsize=14)
# ax.set_ylabel('y', fontsize=14)
# ax.set_zlabel('z', fontsize=14)
#drawFunc(ax)
res10 = []
for i in range (50):
    res10.append(calIntInt(n,ax))
print("The double integration is : "+str(res10)+" .\n")
print("Mean value of "+str(n)+" points : "+str(np.mean(res10))+" .\n")
print("The variance of "+str(n)+" points : "+str(np.var(res10))+" .\n")
print("The standard variance of "+str(n)+" points : "+str(np.std(res10))+" .\n")

print("\n-------------------------------------------------------\n")
```

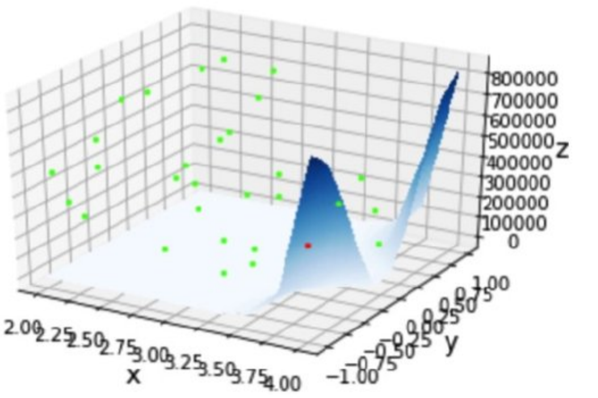
```
-------------------------------------------------------

The double integration is : [0.0, 0.0, 653854.6744944222, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 653854.6744944222, 0.0, 0.0, 0.0, 0.0, 1307
709.3489888443, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 653854.
6744944222, 0.0, 0.0, 0.0, 653854.6744944222, 0.0, 0.0, 0.0, 653854.
6744944222, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0] .

Mean value of 5 points : 91539.65442921911 .

The variance of 5 points : 68575160031.45637 .

The standard variance of 5 points : 261868.5930604439 .


-------------------------------------------------------

The double integration is : [326927.3372472111, 0.0, 0.0, 980782.011
7416332, 0.0, 326927.3372472111, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 326927.3372472111, 0.0, 0.0, 326927.3372472111,
0.0, 0.0, 0.0, 0.0, 0.0, 326927.3372472111, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 326927.3372472111, 0.0, 0.0, 326927.3372472111, 0.0,
0.0, 326927.3372472111, 0.0, 326927.3372472111, 0.0, 0.0, 0.0, 32692
7.3372472111, 0.0, 326927.3372472111] .

Mean value of 10 points : 91539.65442921911 .

The variance of 10 points : 34373085202.79983 .

The standard variance of 10 points : 185399.7982814432 .


-------------------------------------------------------

The double integration is : [326927.3372472111, 163463.66862360554,
0.0, 0.0, 0.0, 490391.0058708166, 0.0, 326927.3372472111, 326927.337
2472111, 0.0, 326927.3372472111, 0.0, 0.0, 326927.3372472111, 0.0,
0.0, 0.0, 163463.66862360554, 163463.66862360554, 0.0, 490391.005870
8166, 163463.66862360554, 0.0, 326927.3372472111, 163463.6686236055
4, 163463.66862360554, 0.0, 163463.66862360554, 0.0, 326927.33724721
11, 0.0, 163463.66862360554, 326927.3372472111, 163463.66862360554,
0.0, 163463.66862360554, 163463.66862360554, 0.0, 0.0, 163463.668623
60554, 0.0, 163463.66862360554, 490391.0058708166, 163463.6686236055
4, 326927.3372472111, 0.0, 326927.3372472111, 163463.66862360554, 32
6927.3372472111, 0.0] .

Mean value of 20 points : 150386.5751337171 .

The variance of 20 points : 23342916070.55809 .

The standard variance of 20 points : 152783.8868158488 .


-------------------------------------------------------

The double integration is : [130770.93489888443, 130770.93489888443,
0.0, 0.0, 0.0, 196156.40234832664, 65385.467449442214, 130770.934898
88443, 65385.467449442214, 261541.86979776886, 65385.467449442214, 6
5385.467449442214, 130770.93489888443, 130770.93489888443, 196156.40
234832664, 0.0, 0.0, 0.0, 130770.93489888443, 0.0, 196156.4023483266
```

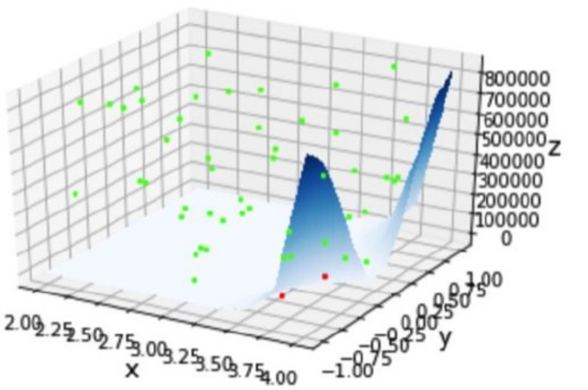
```
4, 65385.467449442214, 261541.86979776886, 130770.93489888443, 13077
0.93489888443, 196156.40234832664, 65385.467449442214, 196156.402348
32664, 196156.40234832664, 0.0, 196156.40234832664, 326927.337247211
1, 0.0, 65385.467449442214, 130770.93489888443, 0.0, 196156.40234832
664, 196156.40234832664, 196156.40234832664, 65385.467449442214, 653
85.467449442214, 65385.467449442214, 0.0, 0.0, 196156.40234832664, 6
5385.467449442214, 196156.40234832664, 65385.467449442214, 130770.93
489888443, 65385.467449442214] .

Mean value of 50 points : 107232.16661708523 .

The variance of 50 points : 7141393224.223485 .

The standard variance of 50 points : 84506.76436962596 .


------------------------------------------------------

The double integration is : [196156.40234832664, 196156.40234832664,
0.0, 65385.467449442214, 98078.20117416332, 130770.93489888443, 9807
8.20117416332, 163463.66862360554, 130770.93489888443, 98078.2011741
6332, 65385.467449442214, 98078.20117416332, 98078.20117416332, 1634
63.66862360554, 32692.733724721107, 196156.40234832664, 32692.733724
721107, 32692.733724721107, 130770.93489888443, 163463.66862360554,
98078.20117416332, 130770.93489888443, 0.0, 196156.40234832664, 2615
41.86979776886, 98078.20117416332, 163463.66862360554, 130770.934898
88443, 196156.40234832664, 65385.467449442214, 130770.93489888443, 6
5385.467449442214, 98078.20117416332, 65385.467449442214, 163463.668
62360554, 98078.20117416332, 163463.66862360554, 163463.66862360554,
196156.40234832664, 98078.20117416332, 130770.93489888443, 130770.93
489888443, 98078.20117416332, 130770.93489888443, 163463.6686236055
4, 163463.66862360554, 98078.20117416332, 0.0, 228849.13607304776, 9
8078.20117416332] .

Mean value of 100 points : 120309.26010697367 .

The variance of 100 points : 3439018624.0214148 .

The standard variance of 100 points : 58643.14643691464 .


------------------------------------------------------

The double integration is : [85001.10768427487, 104616.74791910754,
163463.66862360554, 117693.84140899597, 130770.93489888443, 91539.65
44292191, 91539.6544292191, 124232.3881539402, 143848.02838877286, 1
24232.3881539402, 98078.20117416332, 111155.29466405178, 98078.20117
416332, 91539.6544292191, 150386.5751337171, 130770.93489888443, 143
848.02838877286, 117693.84140899597, 143848.02838877286, 117693.8414
0899597, 130770.93489888443, 170002.21536854975, 111155.29466405178,
104616.74791910754, 111155.29466405178, 150386.5751337171, 98078.201
17416332, 130770.93489888443, 111155.29466405178, 111155.2946640517
8, 111155.29466405178, 91539.6544292191, 143848.02838877286, 91539.6
544292191, 130770.93489888443, 104616.74791910754, 117693.8414089959
7, 124232.3881539402, 124232.3881539402, 111155.29466405178, 170002.
21536854975, 124232.3881539402, 71924.01419438643, 143848.0283887728
6, 111155.29466405178, 143848.02838877286, 124232.3881539402, 98078.
20117416332, 124232.3881539402, 104616.74791910754] .

Mean value of 500 points : 119524.63449758038 .
```

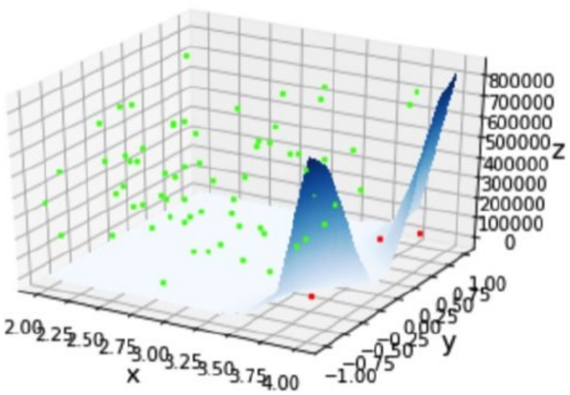
```
The variance of 500 points : 478897451.75084877 .

The standard variance of 500 points : 21883.72572828605 .


------------------------------------------------------

The double integration is : [68654.74082191433, 107886.02129157966,
140578.75501630074, 85001.10768427487, 124232.3881539402, 137309.481
64382865, 98078.20117416332, 88270.38105674699, 91539.6544292191, 13
4040.20827135653, 134040.20827135653, 114424.56803652388, 114424.568
03652388, 111155.29466405178, 98078.20117416332, 107886.02129157966,
127501.66152641231, 120963.11478146809, 147117.30176124498, 124232.3
881539402, 127501.66152641231, 140578.75501630074, 143848.0283887728
6, 107886.02129157966, 104616.74791910754, 101347.47454663544, 8500
1.10768427487, 140578.75501630074, 81731.83431180277, 111155.2946640
5178, 104616.74791910754, 101347.47454663544, 127501.66152641231, 15
0386.5751337171, 91539.6544292191, 117693.84140899597, 98078.2011741
6332, 124232.3881539402, 111155.29466405178, 85001.10768427487, 1176
93.84140899597, 81731.83431180277, 101347.47454663544, 94808.9278016
9121, 120963.11478146809, 71924.01419438643, 140578.75501630074, 101
347.47454663544, 120963.11478146809, 104616.74791910754] .

Mean value of 1000 points : 111743.76387109674 .

The variance of 1000 points : 410719890.8392756 .

The standard variance of 1000 points : 20266.225372261004 .


------------------------------------------------------

The double integration is : [140578.75501630074, 123578.53347944579,
113116.85868753503, 124232.3881539402, 99385.91052315217, 123578.533
47944579, 111809.1493385462, 136001.7722948398, 91539.6544292191, 12
0309.26010697367, 116386.13206000713, 109847.58531506291, 122924.678
80495137, 116386.13206000713, 101347.47454663544, 113770.7133620294
5, 133386.3535968621, 107886.02129157966, 99385.91052315217, 97424.3
464996689, 96770.49182517448, 102001.32922112985, 112463.0040130406
1, 112463.00401304061, 113770.71336202945, 125540.09750292904, 10330
9.0385701187, 117693.84140899597, 128155.51620090673, 132732.4989223
677, 124232.3881539402, 110501.43998955733, 109193.7306405685, 10657
8.3119425908, 100693.61987214102, 127501.66152641231, 119655.4054324
7925, 116386.13206000713, 123578.53347944579, 110501.43998955733, 12
8155.51620090673, 111809.1493385462, 129463.2255498956, 108539.87596
607408, 126193.95217742347, 100693.61987214102, 110501.43998955733,
90885.79975472468, 115078.4227110183, 104616.74791910754] .

Mean value of 5000 points : 114450.72222350366 .

The variance of 5000 points : 135832856.14051282 .

The standard variance of 5000 points : 11654.735352658714 .


------------------------------------------------------

The double integration is : [117693.84140899597, 107886.02129157966,
119001.55075798483, 116713.05939725436, 103962.89324461312, 109193.7
306405685, 115078.4227110183, 113770.71336202945, 122597.7514677041
5, 117693.84140899597, 108866.8033033213, 105924.45726809638, 11278
```

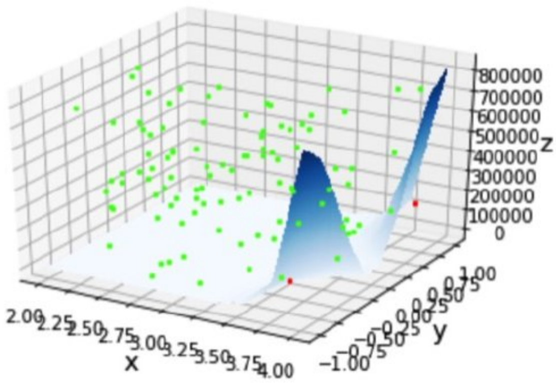
```
9.93135028782, 114424.56803652388, 120963.11478146809, 108866.803303
3213, 119982.33276972648, 122924.67880495137, 106578.3119425908, 121
943.89679320973, 116713.05939725436, 108212.94862882685, 117693.8414
0899597, 114751.49537377109, 115405.3500482655, 111155.29466405178,
110501.43998955733, 113443.78602478224, 126193.95217742347, 120309.2
6010697367, 110501.43998955733, 112789.93135028782, 119655.405432479
25, 117693.84140899597, 111482.22200129897, 108539.87596607408, 1088
66.8033033213, 118347.69608349042, 114751.49537377109, 102001.329221
12985, 113443.78602478224, 114424.56803652388, 106905.23927983802, 1
07232.16661708524, 112789.93135028782, 111809.1493385462, 111809.149
3385462, 111809.1493385462, 111482.22200129897, 118674.62342073761]
.

Mean value of 10000 points : 113724.94353481484 .

The variance of 10000 points : 27720824.40121966 .

The standard variance of 10000 points : 5265.056922885037 .


------------------------------------------------------

The double integration is : [112724.54588283837, 115241.88637964189,
110043.74171741125, 115339.96458081606, 115274.57911336662, 110174.5
1265231014, 111057.2164628776, 109912.97078251235, 116941.9085333274
1, 109684.12164643932, 112070.69120834395, 113509.17149223169, 11370
5.32789458, 113770.71336202945, 114195.71890045084, 111612.992936197
84, 111580.30020247314, 115307.27184709135, 109880.27804878764, 1116
12.99293619784, 112005.3057408945, 116549.59572863075, 112986.087752
63614, 111939.92027344507, 112986.08775263614, 110762.98185935512, 1
13574.55695968113, 113868.79156320362, 112822.62408401254, 109193.73
06405685, 112136.07667579339, 113116.85868753503, 111220.6801315012
1, 112888.00955146198, 114620.7244388722, 117072.67946822628, 11167
8.37840364731, 111809.1493385462, 112103.38394206869, 117105.3722019
51, 112789.93135028782, 111057.2164628776, 113313.01508988337, 11504
5.72997729357, 111972.6130071698, 111024.52372915288, 110632.2109244
5623, 111482.22200129897, 111645.68566992258, 115013.03724356886] .

Mean value of 100000 points : 112761.16174461006 .

The variance of 100000 points : 4003508.478134721 .

The standard variance of 100000 points : 2000.8769272833151 .


------------------------------------------------------
```

The excerpts are shown below.









Experiment results are shown and enhanced experiment results are stored in **data3(_1).xlsx**.

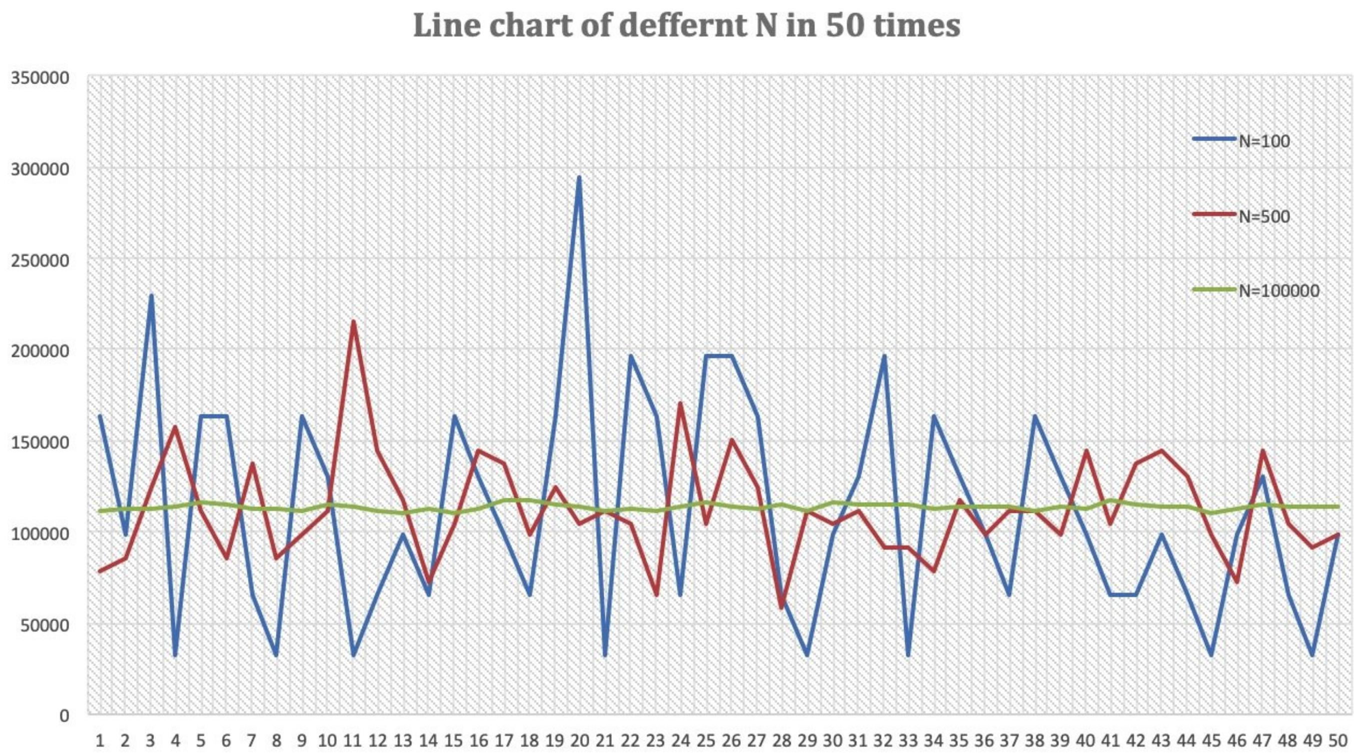
Further Analysis

In [107]:

```
df3 = pd.DataFrame({'Iteration':range(50),'N=100':res5,'N=500':res6,'N=100000':r
es10})
df3.to_excel('data3.xlsx')

N = [5,10,20,50,100,500,1000,5000,10000,100000]

mean = []
mean.append(np.mean(res1))
mean.append(np.mean(res2))
mean.append(np.mean(res3))
mean.append(np.mean(res4))
mean.append(np.mean(res5))
mean.append(np.mean(res6))
mean.append(np.mean(res7))
mean.append(np.mean(res8))
mean.append(np.mean(res9))
mean.append(np.mean(res10))

variance = []
variance.append(np.var(res1))
variance.append(np.var(res2))
variance.append(np.var(res3))
variance.append(np.var(res4))
variance.append(np.var(res5))
variance.append(np.var(res6))
variance.append(np.var(res7))
variance.append(np.var(res8))
variance.append(np.var(res9))
variance.append(np.var(res10))

standard = []
standard.append(np.std(res1))
standard.append(np.std(res2))
standard.append(np.std(res3))
standard.append(np.std(res4))
standard.append(np.std(res5))
standard.append(np.std(res6))
standard.append(np.std(res7))
standard.append(np.std(res8))
standard.append(np.std(res9))
standard.append(np.std(res10))

df3 = pd.DataFrame({'Points':N,'Mean':mean,'Variance':variance,'Standard':standa
rd})
df3.to_excel('data3_1.xlsx')
```

To systematically analyze our experiment results, a great amount of data is of necessity. So we scaled up the experiment and even go so far as to 100000 points in 10 times. Enhanced experiment results are shown in the figure below. The huge amount of data illuminates us some important ideas:

- First, it can be seen that the system variance of calculation become smaller when we conduct our experiment with more random points.
- With the cement data support, we can get the value of result stablizes around 1.1286e+5. It's quite obvious that the Monte Carlo method can be recognized as effective numerical computation to estimate incalculable value.



Points	5	10	20	50	100	500	1000	5000	10000	100000
Mean	91539.65443	91539.65443	150386.5751	107232.1666	120309.2601	119524.6345	111743.7639	114450.7222	113724.9435	112761.1617
Variance	68575160031	34373085203	23342916071	7141393224	3439018624	478897451.8	410719890.8	135832856.1	27720824.4	4003508.478
Standard	261868.5931	185399.7983	152783.8868	84506.76437	58643.14644	21883.72573	20266.22537	11654.73535	5265.056923	2000.876927