CS 3320 – Numerical Software
Module 12 Homework
Monte Carlo Simulation
Solution

1. (10 pt) Evaluate $\iint_R e^{x^2y^2}\,dxdy$ where the region of integration, R, is the unit circle in the *x*-*y* plane. Use 1,000,000 points inside the unit circle to estimate the integral. Use the function average technique or type 1 Monte Carlo simulation.
   Notes:
   a. Generate pairs of random number (x, y) only keep the pairs that satisfy $x^2 + y^2 \leq 1$. You may need more than 1,000,000 pairs to get the required set.
   b. The area of the unit circle is $\iint_R dxdy = \pi$.

   Generate two sequences of random numbers, x and y. Use (x[i], y[i]) to evaluate $e^{x_i^2y_i^2}$. Sum up all the $e^{x_i^2y_i^2}$ and find the average. Multiple the average with π to get your answer. The integral is approximately 3.280

   My Python code is attached here.

```
import numpy as np
import math

def findIntegral(n):
    x=np.zeros(n)
    y=np.zeros(n)

    cnt = -1
    x1 = np.random.uniform(-1,1,n)
    y1 = np.random.uniform(-1,1,n)
    while (1):
        for i in range(n):
            if x1[i]*x1[i]+y1[i]*y1[i] < 1:
                cnt +=1
                x[cnt] = x1[i]
                y[cnt] = y1[i]
                if cnt == (n-1): break
        if cnt == (n-1): break
        x1 = np.random.uniform(-1,1,n)
        y1 = np.random.uniform(-1,1,n)

    funcSum = 0.
    for i in range(n):
        funcSum += math.exp(x[i]*x[i]*y[i]*y[i])

    integral = funcSum*math.pi/n

    return integral
```

2. (10 pt) Find the volume of the 3-d region bounded by the following equations using dart throwing, type 2 Monte Carlo simulation, with 1,000,000 points.

   $0 < x < 1, 0 < y < 1, 0 < z < 1$
   $x^2 + \sin^2 y \le z,$
   $x - z + e^y \le 1.$

   Generate three sequences of random numbers, x, y, and z, in the interval (0, 1). For each pair of (x_i, y_i, z_i), check if $x_i^2 + \sin^2 y_i \le z_i$ and $x_i - z_i + e^{y_i} \le 1$. Count the pairs, cnt, that meet these criteria. Since the volume of the cube is 1, the answer is simply cnt/1000000 which is approximately 0.136.

   You can use pseudo random numbers for x, y, and z. If you use the halton algorithm, it will take longer to generate the random numbers. The code below uses pseudo random numbers to calculate the volume.

```
import numpy as np
import math

def findVol(n):

    x=np.random.uniform(0,1,n)
    y=np.random.uniform(0,1,n)
    z=np.random.uniform(0,1,n)

    count = 0.
    for i in range(n):
        if x[i]*x[i]+math.pow(math.sin(y[i]),2) <= z[i] and \
           x[i]-z[i]+math.exp(y[i])<=1:
            count +=1

    return (float)(count)/n
```