Assignment 5

Chao Cheng

February 28, 2019

1 Question 1

Suppose that you generate standard Gaussian random variables until you have generated N of them such that $S/\sqrt{N} < 0.01$ where S is the sample standard deviation of the N values.

- **a.** How many Gaussian do you think will be generated?
- **b.** How many Gaussian did you generate?
- **c.** What is the sample mean and the sample variance of all the Gaussian generated?

1.1 Answer

a. $E\left[S/\sqrt{N}\right] < 0.01$ Then, we have $N > 10000E(S)$. Since S is not biased, then

$$N > 10000$$

b. I generated 10249 times, see part Code.
c. The sample mean is 0.00764296, the sample variance is 1.02483.

1.2 Code & Output

```
#include<iostream>
#include<cmath>
#define pis 8*atan(1)//2pi
using namespace std;
double boxMuller(){
    double u1, u2;//u(0,1)
    double x, y;  //gsn(0,1)
    double tmp,tmp2;
    u1 = ((double)rand()/(RAND_MAX));
    u2 = ((double)rand()/(RAND_MAX));
    tmp = sqrt(-2*log(u1));
    tmp2 = pis*u2;
    x = tmp * cos(tmp2);
    y = tmp * sin(tmp2);
    return x;
}
int main(){
    int i; // counter realizations
    double tol;    // S/sqrt(N)
    double z, mean, svs;// gsn(0,1), mean, sample variance
    srand(1) ;    // set up random seed
    z = boxMuller();
    mean = z;
    svs = 0;
    i = 1;
    tol = 10;   //initialize error
    while(tol >= 0.0001 ){
        z = boxMuller();
        svs = svs *(i - 1)/i  + (mean - z)*(mean - z)/(i+1);
        mean = mean + (z-mean)/(i+1);
        i = i + 1;
        tol = svs/i;
    }
    cout << "Total Gaussian generated: "<< i << endl;
    cout << "Sample mean is " << mean << endl;
    cout << "Sample variance is " << svs << endl;
}
```

Results:

```
Total Gaussian generated: 10249
Sample mean is 0.00764296
Sample variance is 1.02483
```

2 Question 2

Use Monte Carlo integration to approximate

$$\int_0^1 e^{x^2}\,dx. \tag{1}$$

Simulate until the standard deviation of your estimator of your estimator is less than 0.001.

Estimator of the integral is

$$\frac{1}{N}\sum_{i=1}^{n} e^{u^2}$$

where u is uniformly distributed between (0,1).

2.1 Code

```
#include<iostream>
#include<cmath>
using namespace std;
int main(){
    int i; // counter realizations
    double u;          // u(0,1),
    double x;          //realization of exp(u^2)
    double est;        // estimator(integral)
    double mean, svs;  //mean, sample variance of estimator
    srand(3);          //set up the random seed
    u = ((double)rand()/(RAND_MAX));
    x = exp(u*u);
    est = x ;
    svs = 0;
    mean = est;
    i = 1;
    while(svs >=0.000001 || i<100){
        u = ((double)rand()/(RAND_MAX));
        x = exp(u*u);
        est = est + (x - est)/(i+1);
        svs = svs *(i - 1)/i  + (mean - est)*(mean - est)/(i + 1);
        mean = mean + (est - mean)/(i+1);
        i = i + 1;
    }
    cout<<"Total␣realization␣is␣"<<i<<endl;
    cout<<"The␣integration␣via␣Monte␣Carlo␣simulation␣is␣"<< est <<endl;
}
```

2.2 Results

```
Total realization is 4839335
The integration via Monte Carlo simulation is 1.46253
```

3 Question 3

In Homework 1, you were asked to estimate $E(M)$ where M is equal to the number of uniformly distributed on (0,1) random numbers that must be summed to exceed 1. In other words, for uniformly distributed on (0,1) random variables $U_1, U_2, \cdots, U_n$,

$$M = \min\left\{n : \sum_{i=1}^{n} U_i > 1\right\} \tag{2}$$

You observed that it appeared that the expected value was e. Given 95 percent confidence interval estimates of e when using $N = 10^i$ realizations for $i = 2, \cdots, 6$.

For 95 percent confidence interval, $\alpha = 0.05$, the interval is

$$\bar{X} \pm z_{\alpha/2} S/\sqrt{n}$$

where $\bar{X}$ is the mean of the estimator, S is the standard sample deviation, $z_{\alpha/2} = 1.96$

3.1 Code

```
#include<iostream>
#include<cmath>
using namespace std;
int numGreatOne(){
    double u,sum;//u~(0,1), sum of us
    int i;
    sum = 0;
    i = 0;
    while( sum<= 1){
        u = ((double)rand()/(RAND_MAX));
        sum = sum + u;
        i = i + 1;
    }
    return i;
}
int main(){
    int N[5] = {100,1000,10000,100000,1000000};
    double x;          //realization
    double mean, svs;  //mean, sample variance of estimator
    double std, Nsqrt; // standard deviation
    double l_bound, u_bound;
    srand(3);          //set up the random seed
    for(int j=0;j<5;j++){
        x = numGreatOne();
        mean = x;
        svs = 0;
        for(int i=1;i<N[j];i++){
            x = numGreatOne();
            svs = svs *(i - 1)/i  + (mean - x)*(mean - x)/(i+1);
            mean = mean + (x-mean)/(i+1);
        }
        std = sqrt(svs);
        Nsqrt = sqrt(N[j]);
        l_bound = mean - std * 1.96/Nsqrt;
        u_bound = mean + std * 1.96/Nsqrt;
        cout<<"Total realization is "<< N[j] <<endl;
        cout<<"The estimator is "<<mean<<endl;
        cout<<"The 95 percent confidence intervals is "<<endl;
        cout<<"["<< l_bound<<" , "<<u_bound<<"]"<<"\n"<<endl;
    }
}
```

3.2 Results

```
Total realization is 100
The estimator is 2.78
The 95 percent confidence intervals is
[2.594 , 2.966]

Total realization is 1000
The estimator is 2.72
The 95 percent confidence intervals is
[2.6666 , 2.7734]

Total realization is 10000
The estimator is 2.7168
The 95 percent confidence intervals is
```

```
[2.6997 , 2.7339]

Total realization is 100000
The estimator is 2.71693
The 95 percent confidence intervals is
[2.71151 , 2.72235]

Total realization is 1000000
The estimator is 2.71799
The 95 percent confidence intervals is
[2.71628 , 2.71971]
```

4 Question 4

Consider a sequence of random numbers uniformly distributed on (0,1). Let L denote the first random number that is less than its predecessor. In other words, for uniformly distributed on (0,1) random variables $U_1, U_2, \cdots, U_n$,

$$L = \min\{n : U_1 \leq U_2 \leq \cdots \leq U_{n-1} > U_n\}. \tag{3}$$

It can be shown that E[L]=e. Use simulation to estimate $E[L]$ using $N = 10^i$ realizations for $i = 2, \cdots, 6$. Given 95 percent confidence intervals for each or your estimates.

For 95 percent confidence interval, $\alpha = 0.05$, the interval is

$$\bar{X} \pm z_{\alpha/2} S/\sqrt{n}$$

where $\bar{X}$ is the mean of the estimator, S is the standard sample deviation, $z_{\alpha/2} = 1.96$

4.1 Code

```
#include<iostream>
#include<cmath>
using namespace std;
int numTrail(){
    double u,bigU;//u~(0,1)
    int i;

    u = 0;
    bigU = ((double)rand()/(RAND_MAX));
    i = 1;
    while(u<=bigU){
        u = bigU;
        bigU = ((double)rand()/(RAND_MAX));
        i = i + 1;
    }
    return i;
}
int main(){
    int N[5] = {100,1000,10000,100000,1000000};
    double x;           //realization
    double mean, svs;  //mean, sample variance of estimator
    double std, Nsqrt; // standard deviation
    double l_bound, u_bound;
    srand(3);           //set up the random seed
    for(int j=0;j<5;j++){
        x = numTrail();
        mean = x;
        svs = 0;
        for(int i=1;i<N[j];i++){
            x = numTrail();
            svs = svs *(i - 1)/i  + (mean - x)*(mean - x)/(i+1);
            mean = mean + (x-mean)/(i+1);
        }
        std = sqrt(svs);
        Nsqrt = sqrt(N[j]);
        l_bound = mean - std * 1.96/Nsqrt;
        u_bound = mean + std * 1.96/Nsqrt;
        cout<<"Total realization is "<< N[j] <<endl;
        cout<<"The estimator is "<<mean<<endl;
        cout<<"The 95 percent confidence intervals is "<<endl;
        cout<<"["<< l_bound<<" , "<<u_bound<<"]"<<"\n"<<endl;
    }
}
```

4.2 Results

```
Total realization is 100
The estimator is 2.59
The 95 percent confidence intervals is
[2.44234 , 2.73766]

Total realization is 1000
The estimator is 2.727
The 95 percent confidence intervals is
[2.66888 , 2.78512]

Total realization is 10000
The estimator is 2.729
The 95 percent confidence intervals is
```

```
[2.71182 , 2.74618]

Total realization is 100000
The estimator is 2.71118
The 95 percent confidence intervals is
[2.70578 , 2.71658]

Total realization is 1000000
The estimator is 2.71818
The 95 percent confidence intervals is
[2.71646 , 2.71989]
```

5 Question 5

In Homework 3, you were asked to continually roll a pair of fair dice until all possible outcomes $2, 3, \cdots, 12$ had occurred at least once and conduct a simulation study to approximate the expected number of dice rolls that are needed. Give 95 percent confidence interval estimates of your results when using $N = 10^i$ realizations for $i = 2, \cdots, 6$ in your study.

5.1 Code

```
#include<iostream>
#include<cmath>
using namespace std;
int twoDices(){
    double u1,u2 ; //u1,u2~u(0,1)
    int n1,n2;     //n1,n2~{1,2,3,4,5,6}
    int outcome;   //n1+n2
    int outComes[11] = {2,3,4,5,6,7,8,9,10,11,12}; //outcomes
    int outComeSum;//indicator whether all possible outcomes are shown up
    int const maxSum = 77; // sum of all outcomes
    int i;

    outComeSum = 0;
    i = 0;
    while(outComeSum != maxSum){
        u1 = ((double) rand() / (RAND_MAX));
        u2 = ((double) rand() / (RAND_MAX));
        n1 = (int)(u1 * 6.0) + 1 ;
        n2 = (int)(u2 * 6.0) + 1 ;
        outcome = n1 + n2;
        if( outcome == outComes[outcome-2]){
            outComes[outcome-2] = 0;
            outComeSum = outComeSum + outcome;
        }
        i = i + 1;
    }
    return i;
}
int main(){
    int N[5] = {100,1000,10000,100000,1000000};
    double x;          //realization
    double mean, svs;  //mean, sample variance of estimator
    double std, Nsqrt; // standard deviation
    double l_bound, u_bound;
    srand(3);          //set up the random seed
    for(int j=0;j<5;j++){
        x = twoDices();
        mean = x;
        svs = 0;
        for(int i=1;i<N[j];i++){
            x = twoDices();
            svs = svs *(i - 1)/i  + (mean - x)*(mean - x)/(i+1);
            mean = mean + (x-mean)/(i+1);
        }
        std = sqrt(svs);
        Nsqrt = sqrt(N[j]);
        l_bound = mean - std * 1.96/Nsqrt;
        u_bound = mean + std * 1.96/Nsqrt;
        cout<<"Total realization is "<< N[j] <<endl;
        cout<<"The estimator is "<<mean<<endl;
        cout<<"The 95 percent confidence intervals is "<<endl;
        cout<<"["<< l_bound<<" , "<<u_bound<<"]"<<"\n"<<endl;
    }
}
```

5.2 Results

```
Total realization is 100
The estimator is 62.25
The 95 percent confidence intervals is
[54.8695 , 69.6305]

Total realization is 1000
The estimator is 60.992
The 95 percent confidence intervals is
[58.6301 , 63.3539]
```

```
Total realization is 10000
The estimator is 61.09
The 95 percent confidence intervals is
[60.3927 , 61.7873]

Total realization is 100000
The estimator is 61.0883
The 95 percent confidence intervals is
[60.8677 , 61.309]

Total realization is 1000000
The estimator is 61.1872
The 95 percent confidence intervals is
[61.1169 , 61.2576]
```