NAMES: Zimin Luo, Li Weijie, Adinda, Shang Lin

Calculate the integral.

$$\int_{-1.03}^{0.54} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

Write the maxima code:

```
(%i42) float(integrate(1/sqrt(2*%pi)*%e^((-0.5)*x^2),x,-1.03,0.54));

rat: replaced 1.57 by 157/100 = 1.57
rat: replaced -1.03 by -103/100 = -1.03
rat: replaced 0.54 by 27/50 = 0.54
rat: replaced 1.57 by 157/100 = 1.57
rat: replaced -0.5 by -1/2 = -0.5
rat: replaced -0.5 by -1/2 = -0.5
rat: replaced 1.57 by 157/100 = 1.57
rat: replaced -1.03 by -103/100 = -1.03
rat: replaced 0.54 by 27/50 = 0.54
rat: replaced 1.57 by 157/100 = 1.57
rat: replaced -0.5 by -1/2 = -0.5
rat: replaced -0.5 by -1/2 = -0.5
rat: replaced 1.03 by 103/100 = 1.03
rat: replaced 0.54 by 27/50 = 0.54

(%o42)                          0.5538964809959579
```

The result: 0.5538964809959579

Write the two lines of the R code which you have to modify to calculate the definite integral by MC. [use integrals_MC.R]

```
funkcja3<-function(x) {(2*pi)^(-0.5)*exp(-0.5*x^2)}
integral_1 <- mcint(funkcja3,-1.03,0.54,100000)
```

changes in the original code:

```
mcint<-function(f, a, b, iter){        # definition of the function
  l1<-b-a;
  l2<-l1/1000;
  x<-seq(a,b,by=l2);
  y<-matrix(NA,1001,2);
  y[,1]<-x;
  y[,2]<-f(x);
  plot(y[,1],y[,2],pch=".", col="red");
  lines(y[,1],y[,2],col="red");
  s1<-0;
  s2<-0;
  sup1<-min(min(y[,2]), 0);
  sup2<-max(y[,2]);
  lines(c(a,a),c(sup1,sup2),col="black");
  lines(c(b,b),c(sup1,sup2),col="black");
  lines(c(a,b),c(sup2,sup2),col="black");
  lines(c(a,b),c(sup1,sup1),col="black");
  if(sup2==Inf) integral<-Inf;
  if(sup2!=Inf){
```

```
    for(i in 1:iter){
      m<-runif(1,a,b);

      n<-runif(1,sup1,sup2);
      r<-f(m);
      if(r>0){
        if(n>r | n<0){
          s2<-s2+1;
          points(m,n,pch=".", col="black")
          print("case 1")
        };
        if(n<r & n>0){
          s1<-s1+1;
          points(m,n,pch=".", col="green")
          print("case 2")
        }
      };
      if(r<0){
        if(n<r | n>0){

          s2<-s2+1;
          points(m,n,pch=".", col="black")
          print("case 3")
        };
        if(n>r & n<0){
          s1<-s1+1;
          points(m,n,pch=".", col="green")
          print("case 4")}
      };
    };
    pole <-(b-a)*(sup2-sup1);
    procent<-s1/iter;
    # wartość całki
    integral<-procent*pole;
    lines(y[,1],y[,2],col="red");

  };

  return(integral)

  lines(c(a,b),c(0,0),col="blue", lwd=3);
}
```

Symulate the result for 1000 and 100000 irteration. Results N = 1000 : `0.557442046621107` and N = 100000 : `0.553890702346566`. Compare with the Maxima result. How about the accuracy of the simulation?

```
R N=1000 :          0.557442046621107
R N=100000 :        0.553890702346566
MAXIMA :            0.5538964809959579
```

`It is clear from the above results that the R simulation yields better result as the iteration increases; comparing to the Maxima result, the difference is 0.6401134% and 0.001043272% when N = 1000 and 100000 respectively. In conclusion, the accuracy of the simulation is fairly good.`

Interpret the result. Find on your memory the function and interpretation of the definite integral.

A significant improvement in the previous question is observed by changing the iteration from 1000 to 100000. It is therefore reasonable to assume if the number of iteration approaches infinity, the area will be filled with green and black dots, and the result will be converging to the ‘correct’ value. To verify our hypothesis, we further increase the N and run two additional simulation in R:

N = 200000 : 0.554398037242929
N = 500000 : 0.553755413040869

The respective difference between the Maxima results are:
N = 200000 : -0.09055054%
N = 500000 : 0.02546829%

However, we observe no visible improvement by further increasing the iteration, as compared to Maxima’s result, nor convergence of the result. We think one possible explanation is because the result has almost reached the ‘optimal’ value as N=100000; when N is much smaller the improvement is quite obvious:

N = 2000 : 0.55775521631022 → difference : 0.6966528 %
N = 5000 : 0.556314635740301 → difference : 0. 4365716%
N = 50000 : 0.554197608641897 → difference : 0.05436533%