2015-Fall

Monte Carlo Method

ASSN - 03

Division of liberal arts and sciences
20145071 Junghoon Seo

1. Problem cognition

This problem is to compute a two-variable function which includes a double integral. The function is the following :

$$f(X,Y) = \int_0^1 \int_0^{2\pi} \frac{1}{[1+\frac{(X-r\cos\theta)^2+(Y-r\sin\theta)^2}{1-r^2}]^{1/2}} d\theta dr$$

This function has a symmetric property so the problem can be reduced just to estimate f(X, 0) which is the following :

$$f(X,0) = \int_0^1 \int_0^{2\pi} \frac{1}{[1+\frac{(X-r\cos\theta)^2+(r\sin\theta)^2}{1-r^2}]^{1/2}} d\theta dr$$

Then, an estimation for the double integral by simple MCM is known as the following:

$$F_n = \frac{A}{n}\sum_{i=1}^{n} f(x_i, y_i)H(x_i, y_i),$$

where F_n is a midpoint approximation of double integral, A is the area of the rectangle which encloses integral area, n is the iteration number of MCM, x_i and y_i are x and y value produced randomly by RNG, respectively, and $H(x_i, y_i)$ equals unity if (x, y) is in R and is zero otherwise.

In the view of the problem, the integral area has to be dealt as rectangle. Thus, the function $H(x_i, y_i)$ is always unity and the area A is $(2\pi - 0) * (1-0) = 2\pi$.

2. Code (Programming Language : R)

The parameters X is given as the function works. The output of function is a value of f(X, 0). n is set as 10000000.

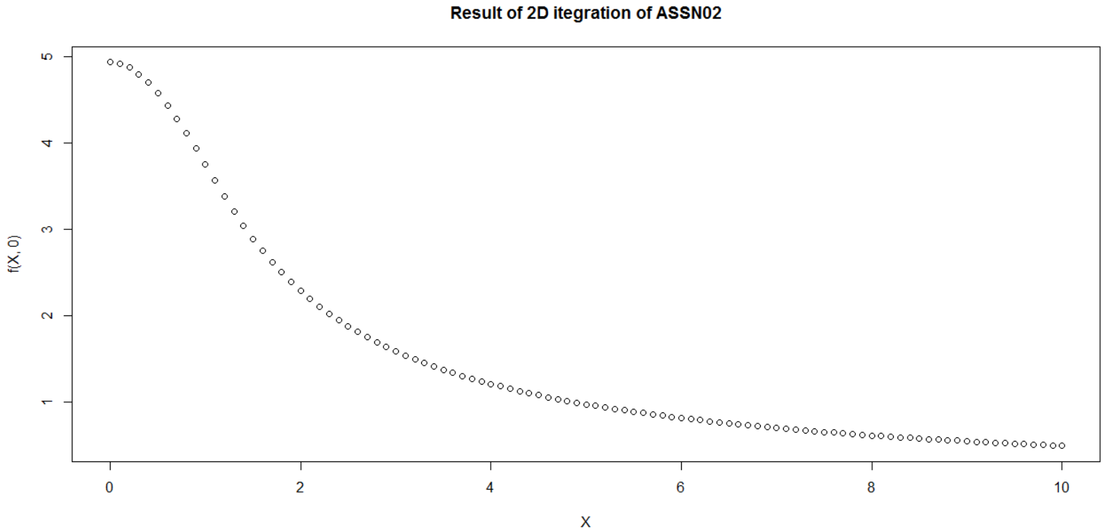
```
particular_integration_function <- function(X){
        n=10000000
        theta_random = runif(n, min=0, max=pi)
        r_random = runif(n, min=0, max=1)
        sum = 0

        for(i in seq(1, n)){
                r = r_random[i]
                theta = theta_random[i]
                sum = sum + 2*pi*(1/(1+((X-r*cos(theta))^2+(r*sin(theta))^2)/(1-r^2))^(1/2))/n
        }
        print(sum)
}
```

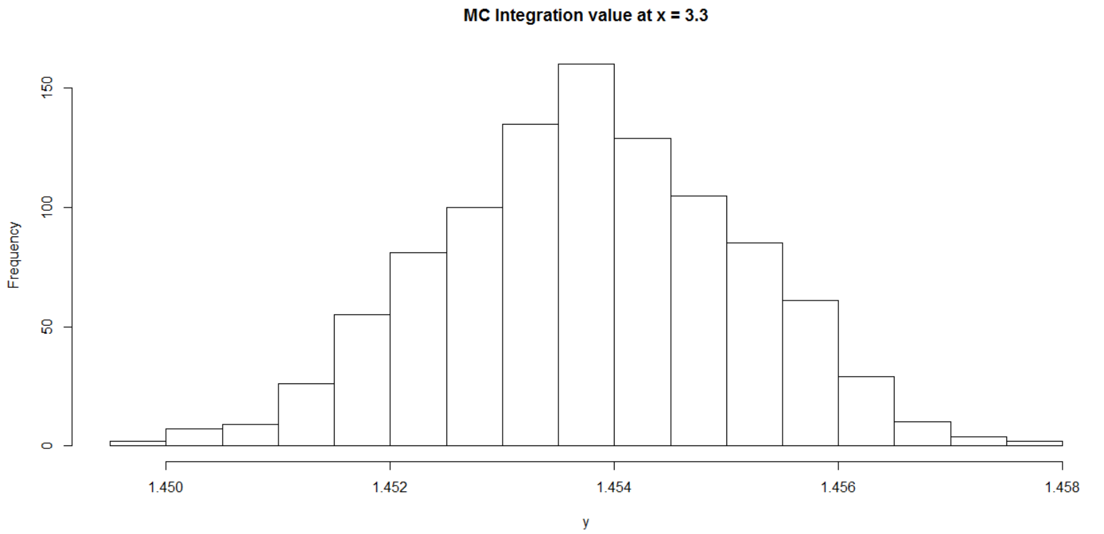
3. Result of implementation

```
> x <- seq(0, 10, 0.1) // sequence X is given as 0.0, 0.1, 0.2, 0.3, ......, 9.9, 10.0
> y<-particular_integration_function(x)
  [1] 4.9339790 4.9186128 4.8731603 4.7994425 4.7001919 4.5786944 4.4384331
  [8] 4.2828011 4.1149061 3.9374642 3.7527770 3.5644929 3.3805814 3.2059236
 [15] 3.0424993 2.8907914 2.7505242 2.6210564 2.5015922 2.3912951 2.2893472
 [22] 2.1949786 2.1074803 2.0262081 1.9505808 1.8800768 1.8142290 1.7526198
 [29] 1.6948757 1.6406623 1.5896805 1.5416618 1.4963653 1.4535741 1.4130928
 [36] 1.3747452 1.3383717 1.3038278 1.2709824 1.2397161 1.2099203 1.1814958
```

```
[43] 1.1543520 1.1284058 1.1035812 1.0798082 1.0570224 1.0351646 1.0141800
[50] 0.9940181 0.9746321 0.9559785 0.9380173 0.9207109 0.9040247 0.8879262
[57] 0.8723853 0.8573738 0.8428653 0.8288353 0.8152606 0.8021196 0.7893922
[64] 0.7770591 0.7651025 0.7535055 0.7422524 0.7313280 0.7207184 0.7104102
[71] 0.7003908 0.6906484 0.6811716 0.6719499 0.6629731 0.6542316 0.6457165
[78] 0.6374189 0.6293308 0.6214444 0.6137522 0.6062472 0.5989227 0.5917723
[85] 0.5847898 0.5779694 0.5713057 0.5647932 0.5584269 0.5522020 0.5461138
[92] 0.5401579 0.5343300 0.5286261 0.5230422 0.5175747 0.5122199 0.5069744
[99] 0.5018349 0.4967982 0.4918613
> plot(x, y, xlab = "X", ylab = "f(X, 0)", main="Result of 2D itegration of ASSN02")
```

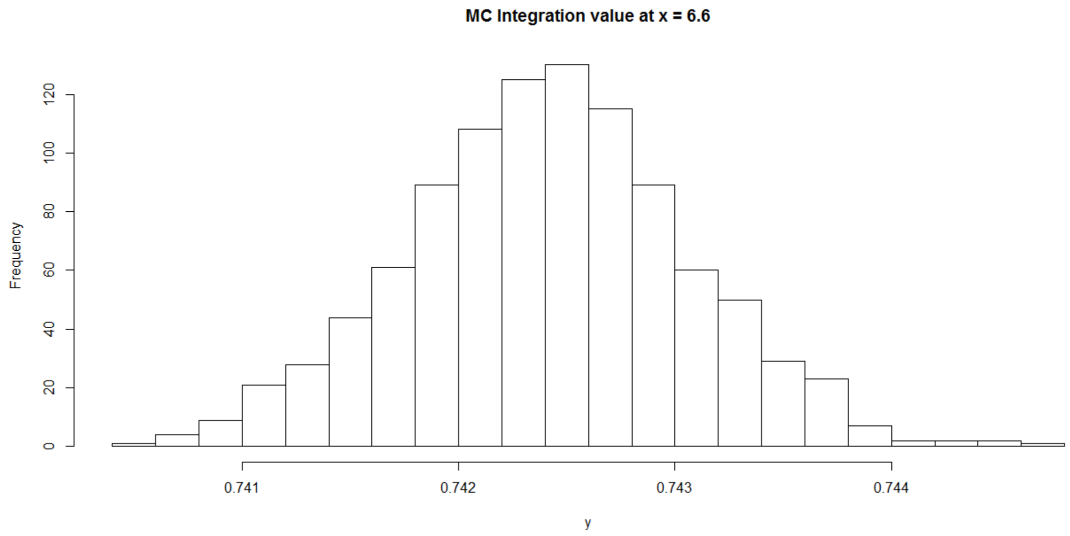



```
> x = 3.3;
> y = c(1:1000);
> for(i in c(1:1000))
+ y[i] = particular_integration_function(x);
> hist(y, breaks=20, main="MC Integration value at x = 3.3")
```




```
> sd(y)
[1] 0.001341782
> mean(y)
[1] 1.453776
> range(y)
[1] 1.449872 1.457693
```

```
> x = 6.6;
> y = c(1:1000);
> for(i in c(1:1000))
+ y[i] = particular_integration_function(x);
> hist(y, breaks=20, main="MC Integration value at x = 6.6")
```




```
> sd(y)
[1] 0.0006465437
> mean(y)
[1] 0.74241
> range(y)
[1] 0.7404809 0.7447702
```

4. Conclusion

It is certain that the result values of MC estimation for integration follow normal distribution by two histograms. The most plausible value of f(3.3, 0) and f(6.6, 0) is respectively proposed as 1.453776, 0.74241 by them and it matches well with the results of the first MC integration. Additionally, when compared to range of result of MC integration, standard deviation of results of it is acceptable.

Unlike FEM, MCM is not entangled in the higher dimension integral, so-called 'the curse of dimension'. The error of MCM by ideal RGN generating random numbers which are perfectly independent is known as $n^{-1/2}$. In the low dimension like 2D, this condition can be accepted approximately. By this basis, the higher dimension integral can be computed by simple MCM, just like the above simple R code.