PROJECT#3 **Monte Carlo Calculations**

Name: Ryota Takei

ID: 995184594

1. Purpose

By using the Monte Carlo Calculations, we try to integrate a function in a region.

2. Introduction

When a function which can be expressed as $f\left(x_1, x_2, x_{3,} \ldots\right) \geq 0$ in a region, we select a point in the region randomly ,$\left(y, x_1, x_2, x_{3,} \ldots\right)$ and get the number of points included (y<f) and excluded (y>f) by f and x-axis, m and n. Then we can integrate the function in the region by the area of the times m/(m+n).

But if f can have negative value, we have to count the number of points included by f and x-axis, when f is negative besides m, k. Then we can get the value of the integral by (m-k)/(m+k+n) times the area of the region.

3. Result

(1) $\int_0^1 \sqrt{\frac{2}{\pi}} e^{-\frac{t^2}{2}} dt$

The actual value is error function of $\frac{1}{\sqrt{2}}$ (0.682689).

Table 1 Monte Carlo method for func. (1).

log(# of points)	the value of func.	Error[%]
1	0.718096	5.186403
2	0.718096	5.186403
3	0.689372	0.978923
4	0.681792	-0.131392
5	0.682838	0.021825
6	0.682655	-0.004980
7	0.682697	0.001172
8	0.682680	-0.001318
9	0.682678	-0.001611
10	0.682679	-0.001465
Actual	0.682689	

At log(# of points)=7, I get the closest value(1.17×10^{-3}% of error).

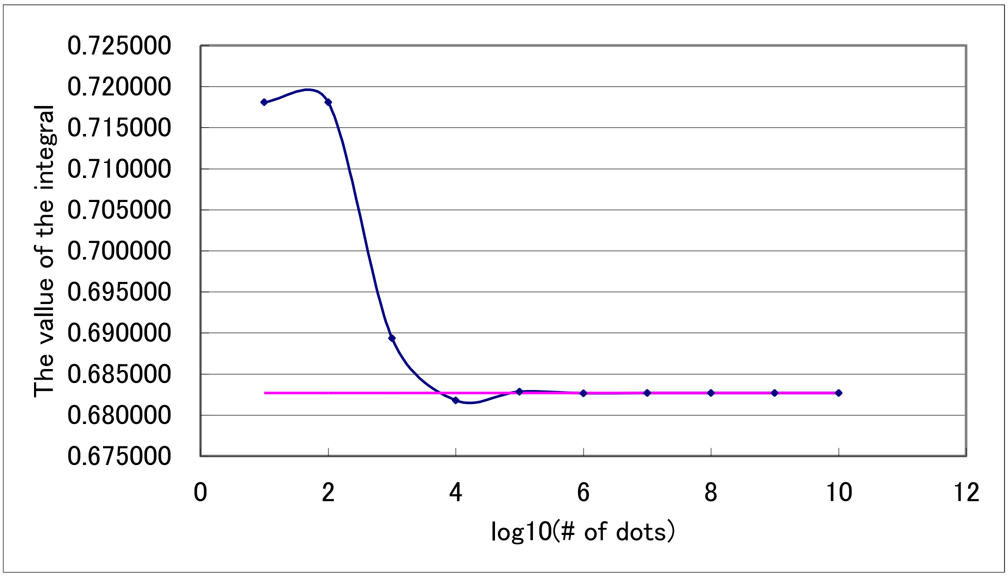

Fig.2 The number of the selected dots and the value of Integral for func. (1)

(2) $\frac{1}{\pi}\int_0^1\int_0^1\int_0^1 e^{-\sqrt{x^2+y^2+z^2}}\,dxdydz$

The actual value is 0.126758.

Table 2 Monte Carlo method for func. (2).

log(# of points)	the value of func.	Error[%]
1	0.127324	0.446520
2	0.124141	-2.064564
3	0.126051	-0.557756
4	0.123663	-2.441660
5	0.126496	-0.206693
6	0.126599	-0.125436
7	0.126778	0.015778
8	0.126758	0.000000
9	0.126756	-0.001578
10	0.126758	0.000000
Actual	0.126758	

I get the exact value at log(# of the points)=8 and 10.

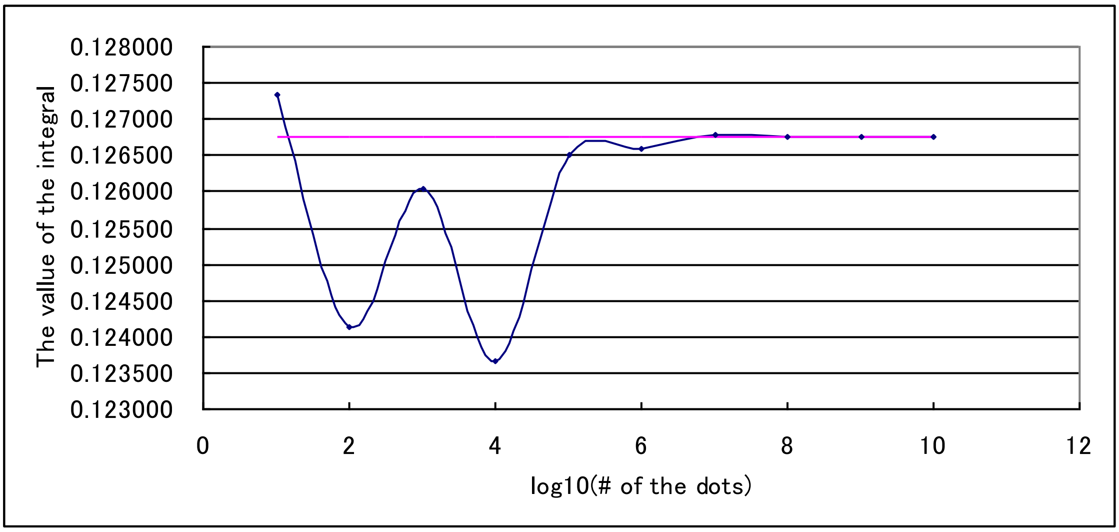

Fig.2 The number of the selected dots and the value of Integral for func. (2)

(3) $\int_0^1 \frac{dx}{\sqrt{1-x^2}}$

The actual value is $\frac{\pi}{2}$=1.570796.

(i) 1^{st} method by using random points

$\frac{1}{\sqrt{1-x^2}}$ becomes infinite as $x \to \infty$, so I have to make approximation, for example, ($0 \le x \le 0.99999$).

Table 3 Monte Carlo 1^{st} method for func. (3).

log(# of points)	the value of func.	Error[%]
1	0.000000	-100.000000
2	4.472102	183.929464
3	1.565236	-0.624669
4	1.498154	-4.883640
5	1.529459	-2.896116
6	1.594081	1.206673
7	1.577534	0.156120
8	1.575502	0.027110
9	1.573696	-0.087551
10	1.573695	-0.087615
Actual	1.575075	

At log(# of points)=8, I get the closest value(2.71×10^{-2} % of error).

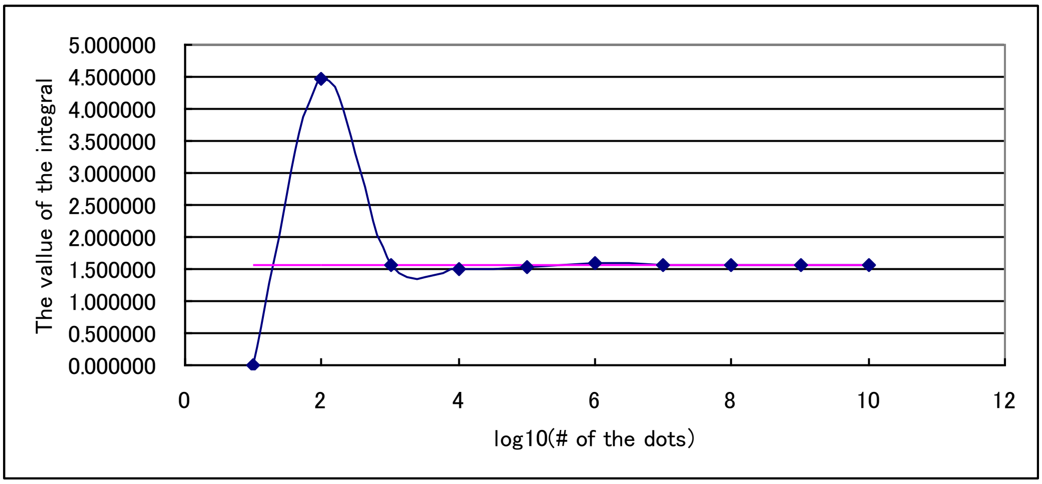

Fig.3 The number of the selected dots and the value of Integral for func. (3) with 1st method

(ii) 2nd method by using random strips

Table 4 Monte Carlo 2nd method for func. (3).

log(# of points)	the value of func.	Error[%]
1	1.300253	-17.448185
2	1.379339	-12.427091
3	1.550431	-1.564624
4	1.533477	-2.641017
5	1.558515	-1.051379
6	1.566357	-0.553497
7	1.565409	-0.613685
8	1.565147	-0.630319
9	1.565139	-0.630827
10	1.565106	-0.632922
Actual	1.575075	

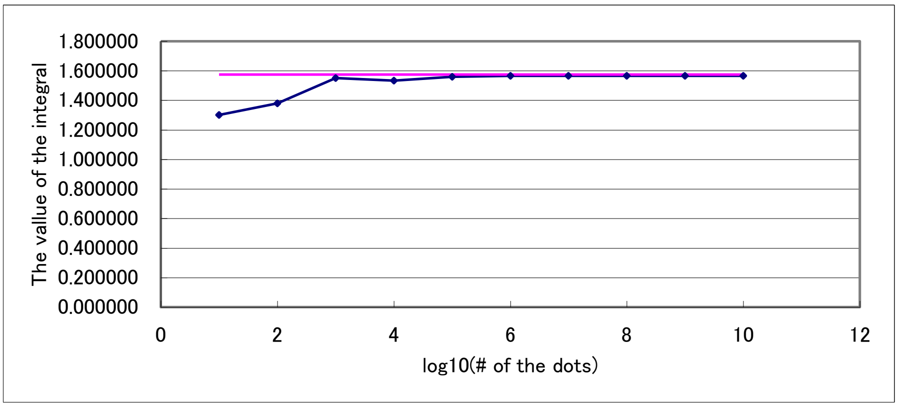

Fig.4 The number of the selected dots and the value of Integral for func. (3) with 2nd method

(4) Random function (rand())

The random function which I used in C++ behaves such as the histogram (Fig.4).

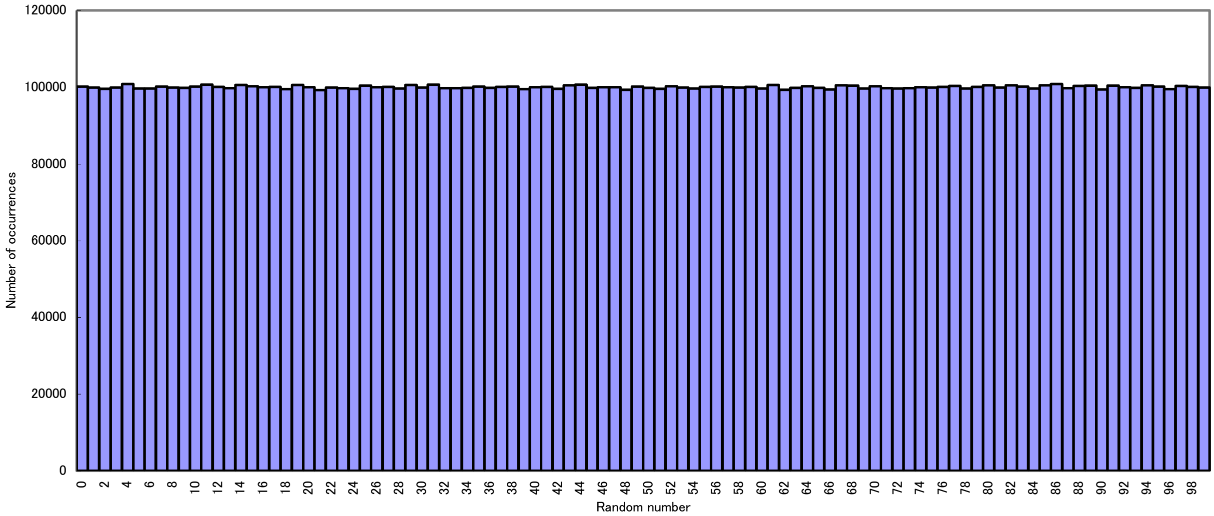

Fig.4 Distribution of 10^8 random numbers generated using rand() in C++.

Number of occurrences is 100,000±352.