Computational Physics – Problem Set 3

Tony Liu B05202068

October 13, 2019

1 Random walk on a two-dimensional square lattice

Let $\vec{x}_N$ be the trajectory of a random walk in two dimensions after N steps. During the lattice random walk process,

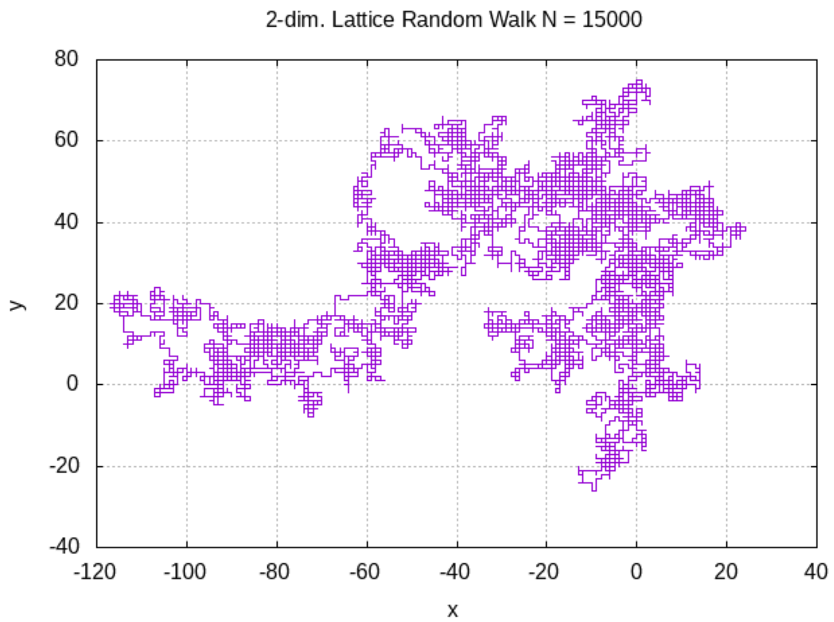
$$\vec{x}_N = (x_N, y_N) = \vec{s}_1 + ... + \vec{s}_N,$$

where $\vec{s}_1, ..., \vec{s}_N$ are independent random vectors with

$$\vec{s}_i = \begin{cases} (1,0), & \text{with probability } \frac{1}{4} \\ (0,1), & \text{with probability } \frac{1}{4} \\ (-1,0), & \text{with probability } \frac{1}{4} \\ (0,-1), & \text{with probability } \frac{1}{4} \end{cases}.$$

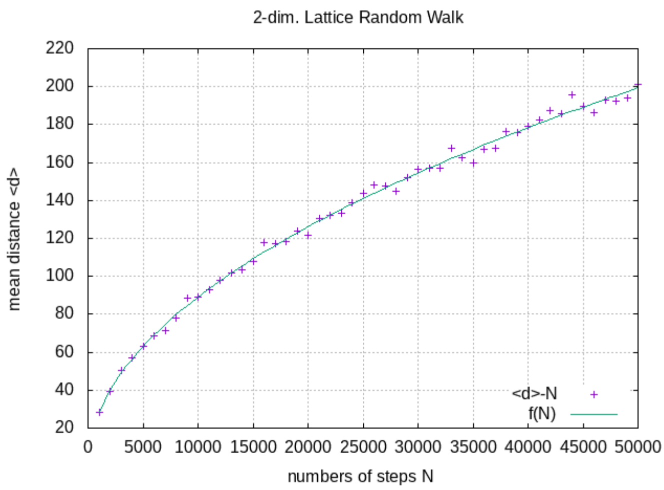
For each N, $\vec{x}_N$ is a two-dimensional vector with components of two integers. The equal probability in all directions is achieved by making the random number into four parts,$0, 0.25, 0.50, 0.75$. Or one can shift the range of random numbers to $[-0.5, 0.5]$ and compare the signess and the greatness to determine the next step. Below is the code fragment to do that:

```
for(int i = 0; i < N; ++i) {
    dx = gsl_rng_uniform(r);    //dy = gsl_rng_uniform(r);
    if (dx >= 0.75)                         x += 1;
    else if (dx >= 0.5 && dx < 0.75 )       x += -1;
    else if (dx >= 0.25 && dx < 0.5)        y += 1;
    else                                    y += -1;
    //if (fabs(dx) > fabs(dy)){
    //  if (dx > 0) x += 1;
    //  else        x += -1;
    //}
    //else{
    //  if (dy > 0) y += 1;
    //  else        y += -1;
    //}
}
```



Above shows the lattice random walk on two dimensions starting from origin after $N = 15000$ steps.

The distance after N steps is defined as $d = \sqrt{x_N^2 + y_N^2}$. At each step N, the mean distance $< d >$ has been repeatedly computed for 100 times. The figure below shows the plot of $< d >$ as function of N.



The fitting function is $f(N) = 0.891696\sqrt{N}$. The theoratical value is $\frac{\sqrt{\pi N}}{2} = 0.886226\sqrt{N}$

The dependence of $< d >$ on $\sqrt{N}$ can be seen from the expectation of

x_N, y_N and $\vec{x}_N^2$:

$$< x_N >= \sum_i^N s_{x,i} P(i) = 0, < y_N >= \sum_i^N s_{y,i} P(i) = 0,$$

since the probability $P(i)$ are equal and $s_{x,i}, s_{y,i}$ are $+1, -1, 0$.
And

$$\begin{aligned} < \vec{x}_N^2 > &=< x_N^2 + y_N^2 >=< \left(\sum_j^N x_j \right)^2 + \left(\sum_j^N y_j \right)^2 > \\ &=< \sum_j \sum_k x_j x_k + \sum_j \sum_k y_j y_k >= \sum_{j,k} < x_j x_k + y_j y_k > \\ &= 2N + \sum_{j \neq k} < x_j x_k + y_j y_k > . \end{aligned}$$

Here if $j \neq k$, then $x_j x_k$ is equally likely to be $+1, -1$ and 0, so $< x_j x_k >= 0$. Same holds for the y coordinate. Hence, the mean distance $< d >=< |x_N| >$ should be of the order of $\sqrt{N}$. In fact, in general, for $k-$dimensional random walk,

$$< d_k >= \sqrt{\frac{2N}{k}} \frac{\Gamma(\frac{k+1}{2})}{\Gamma(\frac{k}{2})}.$$

2 Monte Carlo integration in 10 dimensions

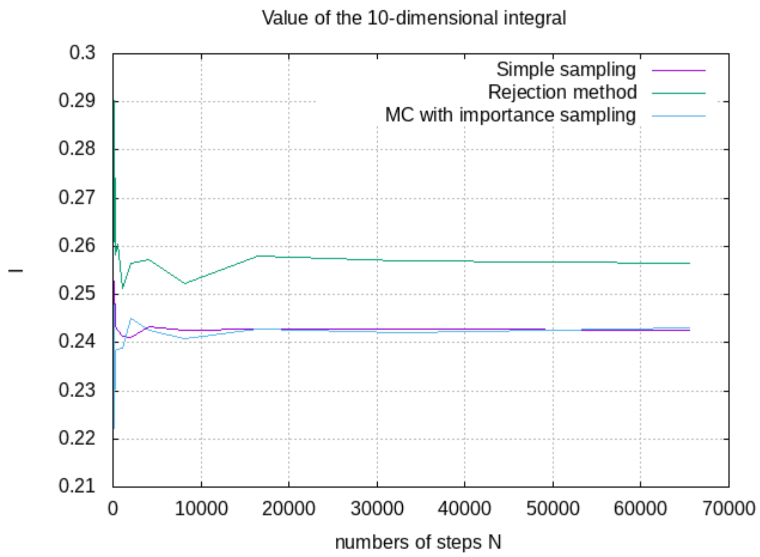
To compute the 10 dimensional integral

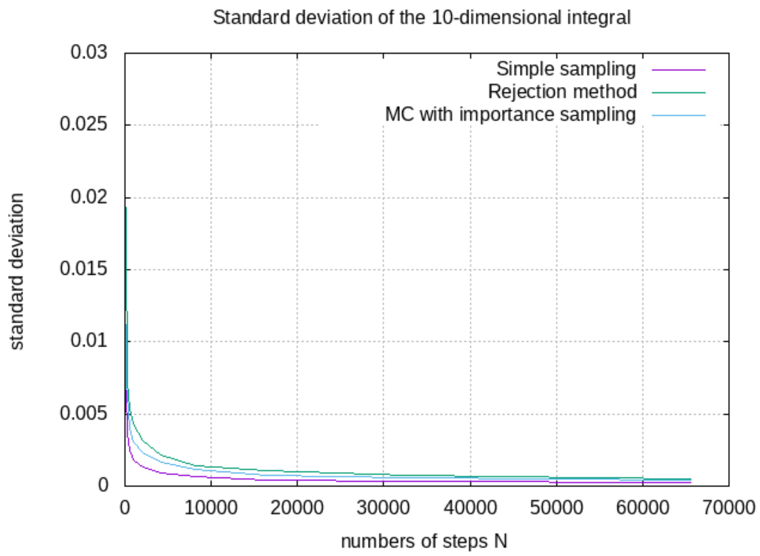
$$I = \int_0^1 dx_1 ... \int_0^1 dx_{10} \frac{1}{1 + x_1^2 + ... + x_{10}^2}$$

Using three different methods, the following table shows the value of I. for different steps:

N	Simple sampling	Rejection method	MC with importance sampling
2	0.265326	0.258097	0.253435
4	0.220418	0.279326	0.211414
8	0.264640	0.274518	0.227476
16	0.233788	0.246127	0.259344
32	0.231881	0.214452	0.273208
64	0.238286	0.231305	0.273650
128	0.252807	0.290414	0.222072
256	0.243330	0.258170	0.238384
512	0.242479	0.260494	0.238536
1024	0.241410	0.251155	0.238845
2048	0.241019	0.256440	0.245110
4096	0.243317	0.257335	0.242564
8192	0.242576	0.252246	0.240708
16384	0.242801	0.257994	0.242834
32768	0.242889	0.256966	0.242093
65536	0.242480	0.256560	0.242960

By using Methamatica, the exact value of I is 0.242853. The one using Motropolis algorithm gives the best estimation of I.




Standard deviation of the 10-dimensional integral
Simple sampling
Rejection method
MC with importance sampling
standard deviation
0.03
0.025
0.02
0.015
0.01
0.005
0
0
10000
20000
30000
40000
50000
60000
70000
numbers of steps N