# Lab 4:
# Random Walks

Corey Mutnik
*Computational Physics 305, University of Hawaii at Manoa*

February 10, 2015

**Abstract**

The problem of random walks crops up throughout all of life. A prime example is brownian motion: the motion of particles suspended in a solution. This motion seems random to an observer. Upon further inspection one realizes this motion to be a result of the forces acting on each particle. Gravitational attraction pulls each particle towards the center of our planet. Various other forces, such as the Coulomb force between particles, also play a role in the overall motion that is observed. Just like the motion itself, computers are unable to generate genuinely random results. Using various commands allows for the generation of pseudo-random numbers. This is done using an algorithm that produces results based on the inital input, also called the 'seed'. Each apparently random number generated by our program will always be generated if the same seed number is used.

## 1 Introduction

This weeks lab pertained to the apparent random motion of various things in everyday life. Since computers are deterministic machines, they are unable to generate truly random numbers. Rather, we use commands such as 'drand48()' to have the computer generate a pseudo-random number. Starting from a common seed value will always output the same results. Therefore, results are reproducable rather than truly random.

## 2 Code

The code for this weeks can be seen online:
http://www2.hawaii.edu/ cmutnik/lab4.html

## 3 Computational problem

Initally the program generated an uncorrelated set of coordinates. The independence of these x, y pairs was resolved by first defining a randomly generated angle that would be used in the computation of both variables. We also bounded the motion from one step to the next by restricting each jump to a length of one unit step. The number of steps taken was graphed against the distance each generated coordinate was from the origin.

### 3.1 Relevant equations

The errors used in generating figures 1-6 were generated using:

$$\frac{r[i]\sqrt{Ntrials}}{(Ntrials)^2} \tag{1}$$

## 4 Graphs

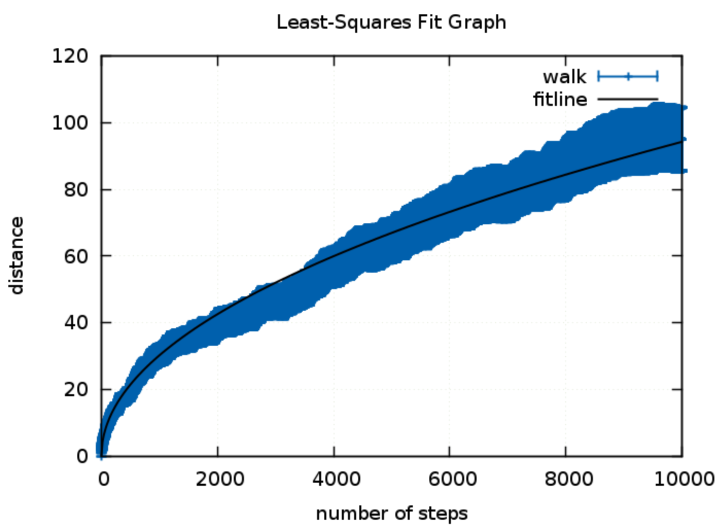


Figure 1: *Random Path using Ntrials=100 and Nmax=10,000*

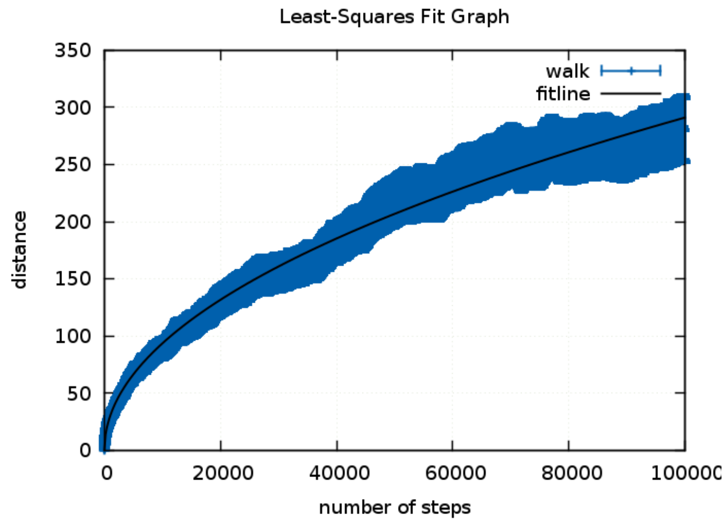


Figure 2: *Random Path using Ntrials=100 and Nmax=100,000*

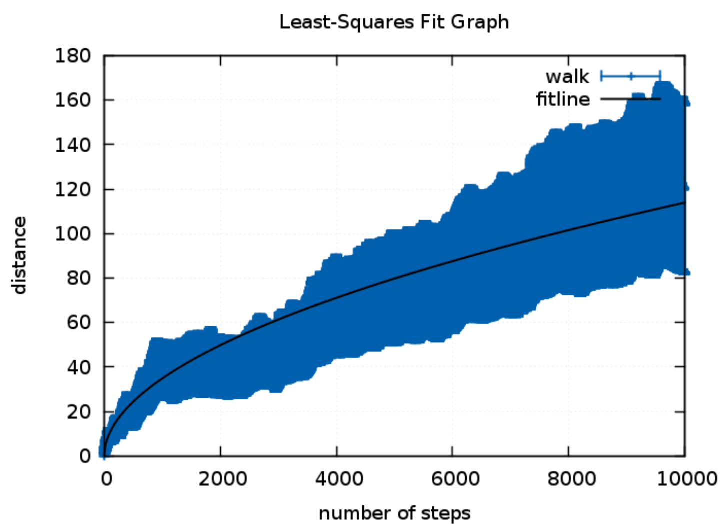


Figure 3: *Random Path using Ntrials=10 and Nmax=10,000*

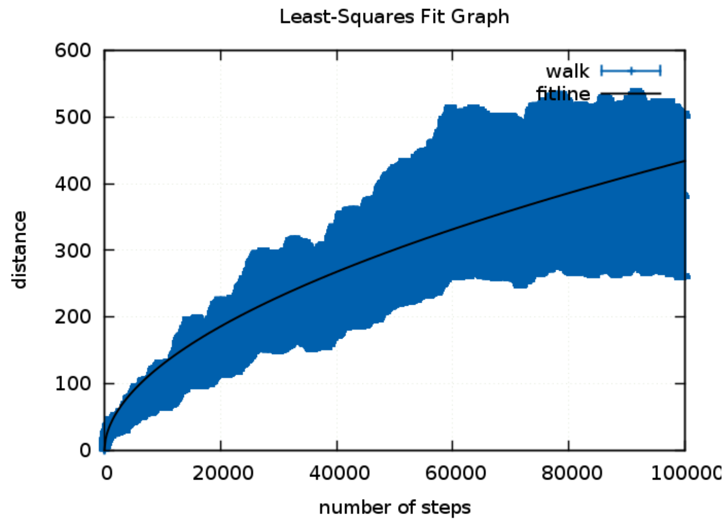


Figure 4: *Random Path using Ntrials=10 and Nmax=100,000*

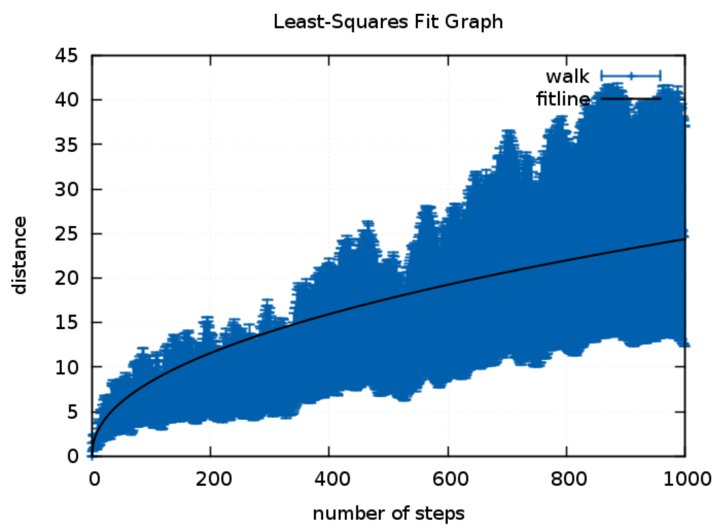


Figure 5: *Random Path using Ntrials=4 and Nmax=1,000*

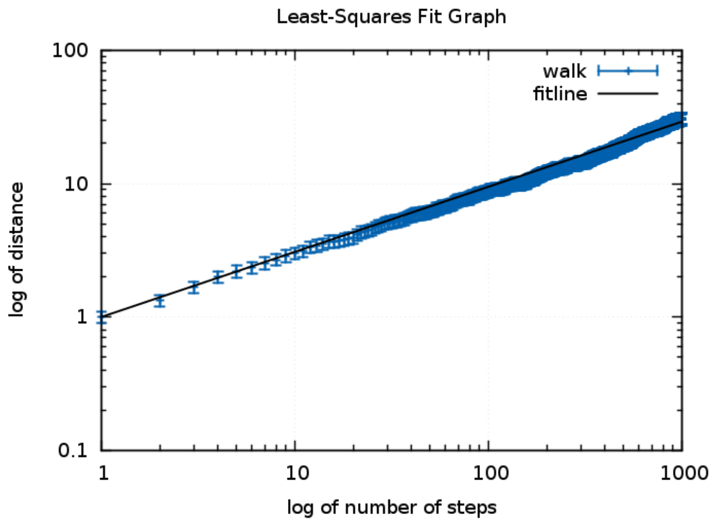


Figure 6: *Random Path plotted on a logrithmic scale using Ntrials=100 and Nmax=10,000*

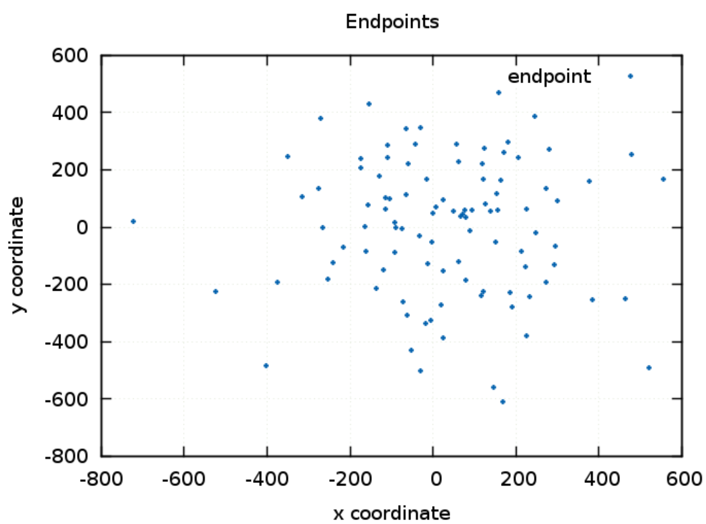


Figure 7: *Endpoints using Ntrials=100 and Nmax=10,000*

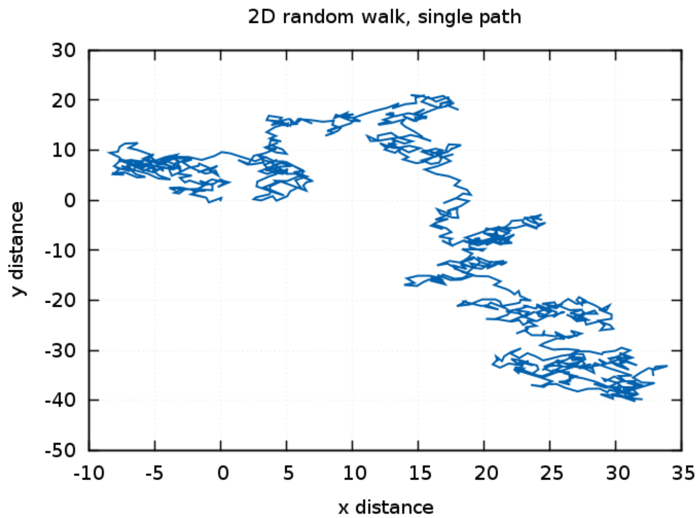


Figure 8: *Path of two dimensional random walk*

# 5 Analysis

Figures 1-6 showed the number of steps graphed against the distance each new coordinate was from the origin, along with a best fit line. Both reducing the number of trials and increasing the maximum number of steps causes the error to increase. This is demonstrated by equation 1 and shown by compairing various figures. The fitted exponent converges to 0.5 at a quicker rate, as the number of trials increases.

Figure 7 is a scatter plot of the endpoints of each trial in two dimensions. As expected, this graph has a distribution that is most dense at the original, approaching an overall even distribution. Since each unit step was taken using a random angle in any direction some steps would be in the opposite direction of the overall motion. These steps, occuring less frequently than those that followed the trend, would cause a decrease in the distance from the origin from one step to the next. For this reason it is expected that the uniformity of the scatter plot would mimic figure 7.

# 6 Conclusion

It may be impossible for computers to generate truly random numbers but using large prime seeds allowed this program to output coordinates with seemingly zero coorelation to each subsequent set. The generation of pseudo-random numbers is extremely benefitial in modeling certian behaviors. There was a significant difference between how long it took to compile the data from paths containing different Ntrial values. As the number of steps increased so did the time necessary to compute and collect the data.

## 7 Extra Credit

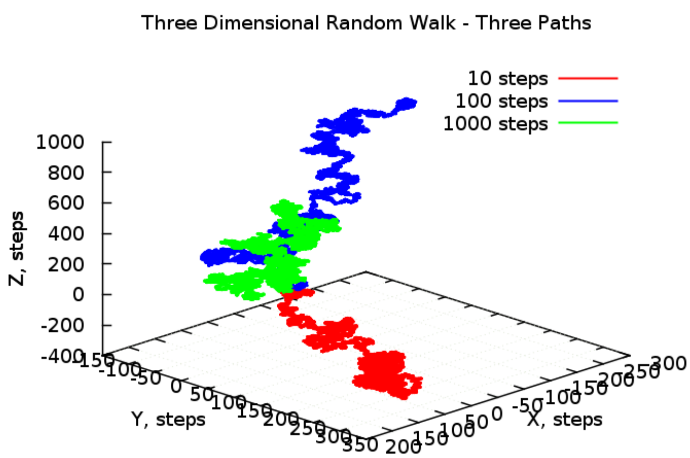


Figure 9: *Three Dimensional Random Walk for three paths - 100, 1,000, and 10,000 steps*

Figure 8 shows the walk path in 2 dimensions for a single path. Figure 9 shows the 3-dimensional walk path for three seperate paths. The path in red represents one containing 100 steps. The path in blue represents one containing 1,000 steps. The path in green represents one containing 10,000 steps.

## References

[1] R. H. Landau and M. J. Paez, "Computational Physics, Problem Solving with Computers," (Wiley: New York) 1997.

[2] P. Gorham, (2014).