Computational Physics

Project 6: Random Walks

Due: March 21st 2018

George Davila[1]*

1 Department of Physics, University of Central Florida, 4111 Libra Drive, Physical Sciences Bldg. 430, 32826, Orlando, United States

Abstract: Here we discuss project #6. We review the context of the problem, its relation to real physical systems, and the methodology by which we tackled its solution. All figures are appended at the end of the paper (pg. 6 - pg. 21).

1. Introduction

Random walks and related phenomena are very useful tools in statistical physics. While an object might be entirely subject to classical mechanics—and thus deterministic (as opposed to objects in quantum mechanics)—it might be the case that there are too many calculations to do. So to model such an objects behavior we can have it undergo some sort of random motion, or random walk, in which the preference for either direction might be weighted or simply eliminated.

Although random walks are today mostly done on computers, they have been thoroughly discussed and analyzed by physicists since the 19th century. They form much of the basis of statistical mechanics and show up in phenomena such as Brownian motion. Such phenomena are so deeply rooted in the history of physics that even Albert Einstein's PhD thesis A New Determination of Molecular Dimensions was based on the concept of Brownian motion. In this work Einstein took the phenomena of Brownian motion—known for nearly a century prior—and gave it a statistical physics explanation, for the first time allowing atoms to be assigned actual masses and volumes. Prior to this it was thought that atoms were the cause of Brownian motion, but it was not known how to use that knowledge to assign them measurable and useful quantities in such a setting.

For the purposes of this project we will deal with random walks in two main cases of distinction to be discussed in the following section.

* *E-mail: GDavila@knights.ucf.edu*

2. Analytics

Our two main cases are distinguished by their time-dependence and the nature of their walks. We discuss the specifics of these in the computational section. Here we describe the general analytical behavior, applicable to all the cases.

All cases consider a system somewhat analogous to a cup of coffee (hence the name of the files). Multiple particles are placed in a volume and allowed to undergo random walks. For this type of system, density is a good quantity to look at. While we can't say much about individual trajectories, we can make predictions about the density of the system in general time regimes. At the beginning, in the low time regime, it should hover around the initial density for a bit while slowly dissipating. As the walkers have more time to move about the system will become more and more diluted and the density should tend to 0. We recover these behaviors in the equations below.

Lets review—very briefly—some of the fundamental theory following Einstein's 1905 paper [4] (Einstein uses f in the paper but we use ρ below) . We can expand the density as

$$\rho(x,t) + \tau\frac{\partial\rho(x)}{\partial t} + \cdots = \rho(x,t+\tau) = \int_{-\infty}^{+\infty} \rho(x+\Delta,t)\cdot\varphi(\Delta)\,\mathrm{d}\Delta \tag{1}$$

$$\rho(x,t) + \tau\frac{\partial\rho(x)}{\partial t} + \cdots = \rho(x,t)\cdot\int_{-\infty}^{+\infty}\varphi(\Delta)\,d\Delta + \frac{\partial\rho}{\partial x}\cdot\int_{-\infty}^{+\infty}\Delta\cdot\varphi(\Delta)\,\mathrm{d}\Delta \tag{2}$$

$$\rho(x,t) + \tau\frac{\partial\rho(x)}{\partial t} + \cdots = \rho(x,t) + \frac{\partial^2\rho}{\partial x^2}\cdot\int_{-\infty}^{+\infty}\frac{\Delta^2}{2}\cdot\varphi(\Delta)\,\mathrm{d}\Delta + \cdots \tag{3}$$

Rearrange the last equation as

$$\tau\frac{\partial\rho(x)}{\partial t} = \frac{\partial^2\rho}{\partial x^2}\cdot\int_{-\infty}^{+\infty}\frac{\Delta^2}{2}\cdot\varphi(\Delta)\,\mathrm{d}\Delta + (\text{higher order terms}) \tag{4}$$

Represent the remaining integral as some function D (it will soon be apparent why we choose this symbol):

$$\tau\frac{\partial\rho(x)}{\partial t} = \frac{\partial^2\rho}{\partial x^2}\cdot D + (\text{higher order terms}) \tag{5}$$

Minus the higher order terms this is not simply a Diffusion equation with a diffusion coefficient D. And if we take D to be constant, it simply becomes the heat equation [5]. Then, starting from a time $t = 0$, the density of the system will evolve in a Gaussian fashion:

$$\rho(x,t) = \frac{N}{\sqrt{4\pi Dt}}e^{-\frac{x^2}{4Dt}} \tag{6}$$

Take the large time limit

$$\rho(x, t\to\infty) = 0 \tag{7}$$

Moreover, in the low time regime, the density decays fairly slowly. So the expected behaviors are recovered. The crossover regime is $x^2 \sim 4Dt$, so in order to prevent the system from dispersing too quickly we could make D very small.

This tells us much about how we might expect our coffee equation to behave. We should expect a fairly symmetric and fairly Gaussian diffusion of our simulated particles. But it is important to note that our systems use a relatively small number of particles $N \sim 10^2$, so they should be expected follow these rules to a rough degree, but by no means exactly. Of course, if we ran an $N \sim 10^{23}$ simulation we would almost certainly see a near-exact adherence to any degree of error we might be reasonably concerned with.

3. Computations

The codes used & the simulation plots produced are shown in the Figures appended.

3.1. Static Case: No Walk

Properties:

1. No time-dependence
2. No walk

This plot is shown in Fig. 3. This is a fairly uninteresting case. We basically just expect it to be an arrangement of the particles on the initial points. This primarily serves as a case to which we can compare the time-dependent cases.

3.2. 2D Time-Dependent Random Walk

While time-dependent, this code also includes a random walk. So we include a simulation with no time steps ($nt = 0$) as well.

Properties:

1. Time-dependence - Varying numbers of times steps
2. Random walk

We use & show in the figures appended a variety of time steps:

1. $nt = 0$ - Fig. 4
2. $nt = 10$ - Fig. 5

3. $nt = 100$ - Fig. 6

4. $nt = 1,000$ - Fig. 7

5. $nt = 10,000$ - Fig. 8

6. $nt = 50,000$ - Fig. 9

7. $nt = 100,000$ - Fig. 10

8. $nt = 0$ - Fig. 11 ($nt = 0$ case rescaled to have same scale as Fig.10)

3.3. 1D Time-Dependent Random Walk

We used a number of walkers $nw = 100$ for both walks and changed the number of time steps. The essential behavior of both sets of walks is the same. The Mean square deviation is somewhat linear in both.

1. 1D random walk for $nt = 100$ - Fig. 12

2. 1D random walk for $nt = 1,000$ - Fig. 13

3. Mean squared deviation of 1D random walk for $nt = 100$ - Fig. 14

4. Mean squared deviation of 1D random walk for $nt = 1,000$ - Fig. 15

4. Discussion & Conclusions

Looking at the simulations in order is is easy to see some general patterns.

- The first few plots, $nt = 10, 10^2, 10^3$ largely retain the pixelated grid structure (and thus much of the density) it is only for the $nt = 10^4$ case that most of the original structure has disappeared. From this fact we might estimate that the crossover time is very roughly of the order $nt \sim 10^4$, at least as far as the structure is concerned.

- The dissipation occurs fairly symmetrically, much as we would expect for a Gaussian heat-like dissipation. There are a few outliers, but even in the $nt = 10^5$ case we can see a decently symmetric diffusion.

- The density goes to 0 and at an increasing rate. Note that the $nt = 10^4, 5 \cdot 10^4, 10^5$ cases are all rescaled. Whereas we began with an initial particle density of 100 particle per 100 units area, we end up with a $nt = 10^5$ particle density of less than 100 per 10,000 units area.

- The systems by no means matched the equations precisely, but they followed the same general behaviors. If we were to look at systems of greater particle number the system would likely follow the equations more closely.

- The mean squared deviation of the 1D walk grows approximately linearly. Random walks are unbiased, so this implies that the variance grows in a linear fashion.

The last item above is of particular analytical note, as the variance can be said to obey the relation

$$msd - (bias^2 = 0) = Variance = 2Dt \tag{8}$$

from the normal distribution relation

$$\mathrm{Var}(X) = \int_{-\infty}^{\infty} \frac{x^2}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx - \mu^2 = \sigma^2 \tag{9}$$

This suggests that D, the effective diffusion coefficient of our system is fairly linear (and thus fairly close to obeying the heat equation). In more thorough analyses, this fact would allow for very useful simplifications. It suggests that we might be able to model systems far larger than those we can simulate using the heat equation. It further suggests that (ideal) heat diffusion and random walk diffusion are very closely related. Of course we already knew them to be, but this provides some numerical evidence to back up the theory.

Thus our expectations were met. And the analysis provided by Einstein hold true (unsurprisingly).

References

[1] S. Stolbov, Computational Physics Class Notes

[2] B. Ydri, *Computational Physics: An Introduction to Monte Carlo Simulations of Matrix Field Theory*, arXiv:1506.02567 [hep-lat] (June 2015)

[3] K. Anagnostopoulos, *Computational Physics Volume 2: A Practical Introduction to Computational Physics and Scientific Computing* pg. 91 (Aug. 2014)

[4] A. Einstein, "On the movement of small particles suspended in a stationary liquid demanded by the molecular-kinetic theory of heat" PhD thesis (1905). 1956 translation by R. Furth & A.D. Cowper.

[5] Weisstein, Eric W. *Heat Conduction Equation.* From MathWorld–A Wolfram Web Resource.

Figure 1. **One of the codes used, entitled *coffee_notimestep.f*, that we wrote in class.**

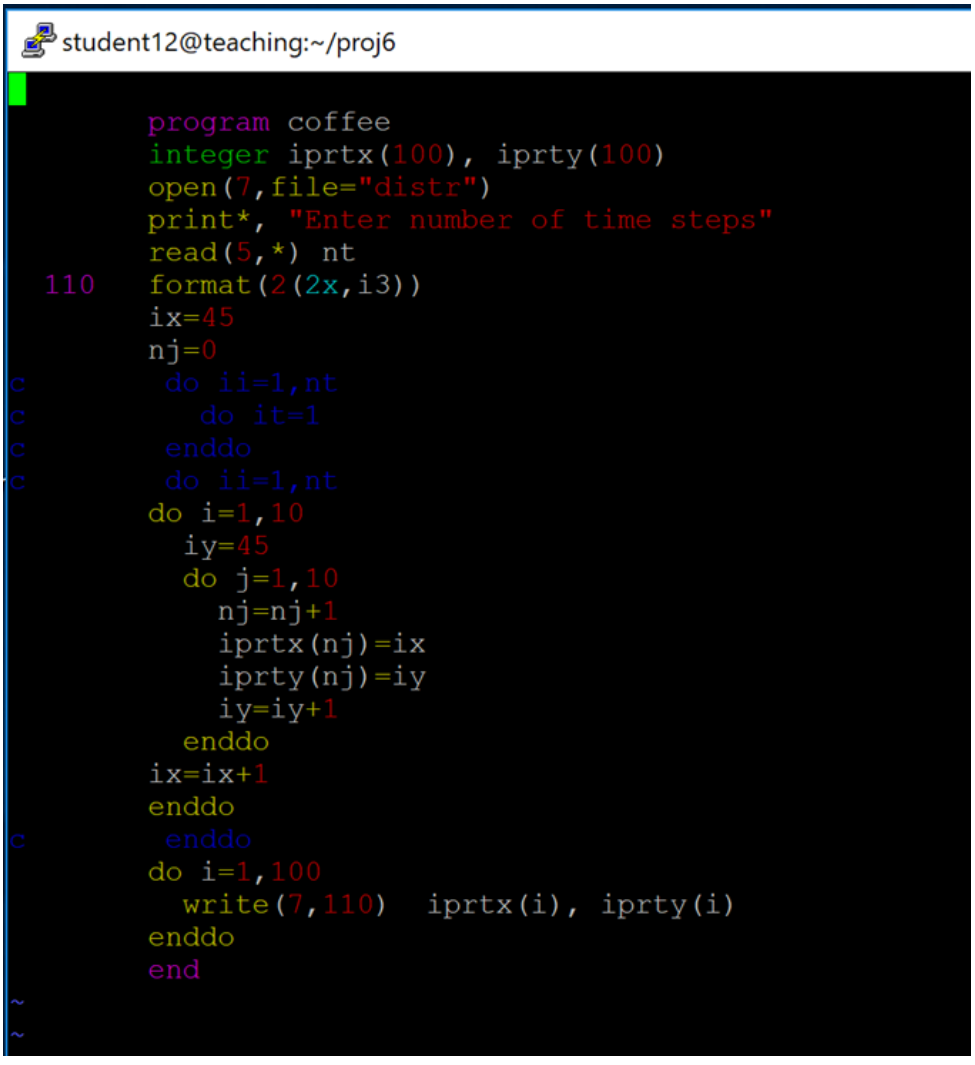

Figure 2. One of the codes used, entitled *coffee.f*, that we wrote in class. This code uses random walks, which can be seen by the inclusion of the *rand* function. The other code *coffee_notimestep.f* has no such pseudo-random number generating function in it.

Figure 3. **Simulation for static no walk case.**

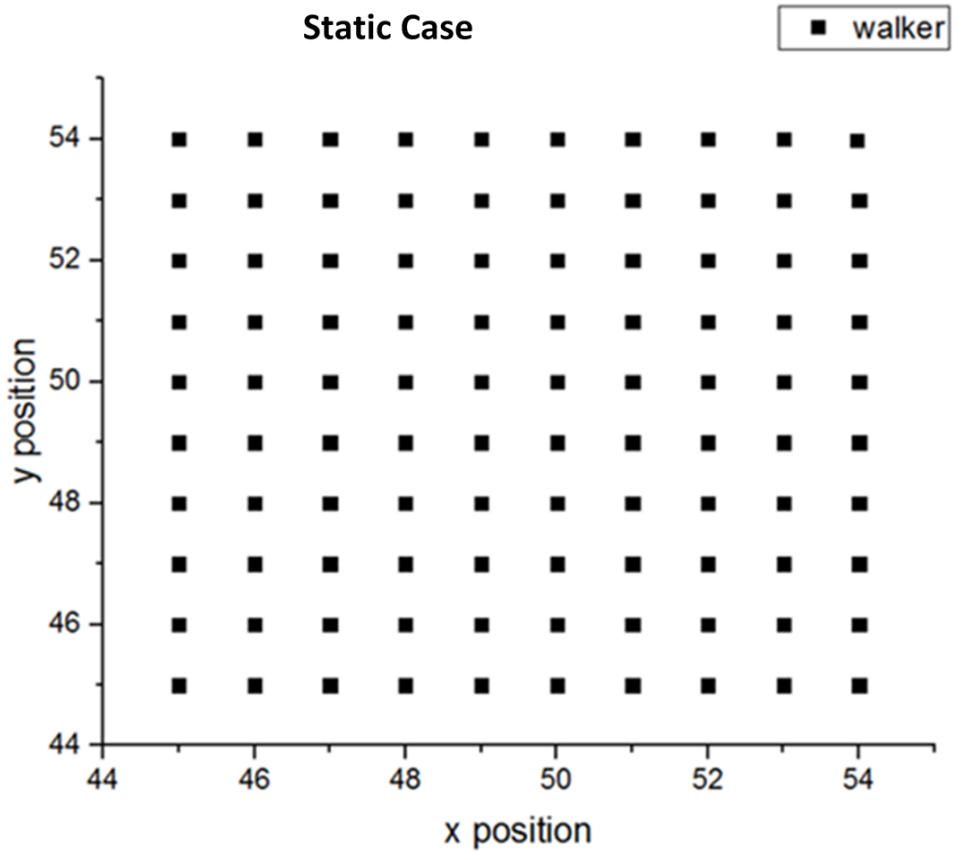

Figure 4. **Simulation for time-dependent random walk case. Number of time steps is $nt = 0$.**

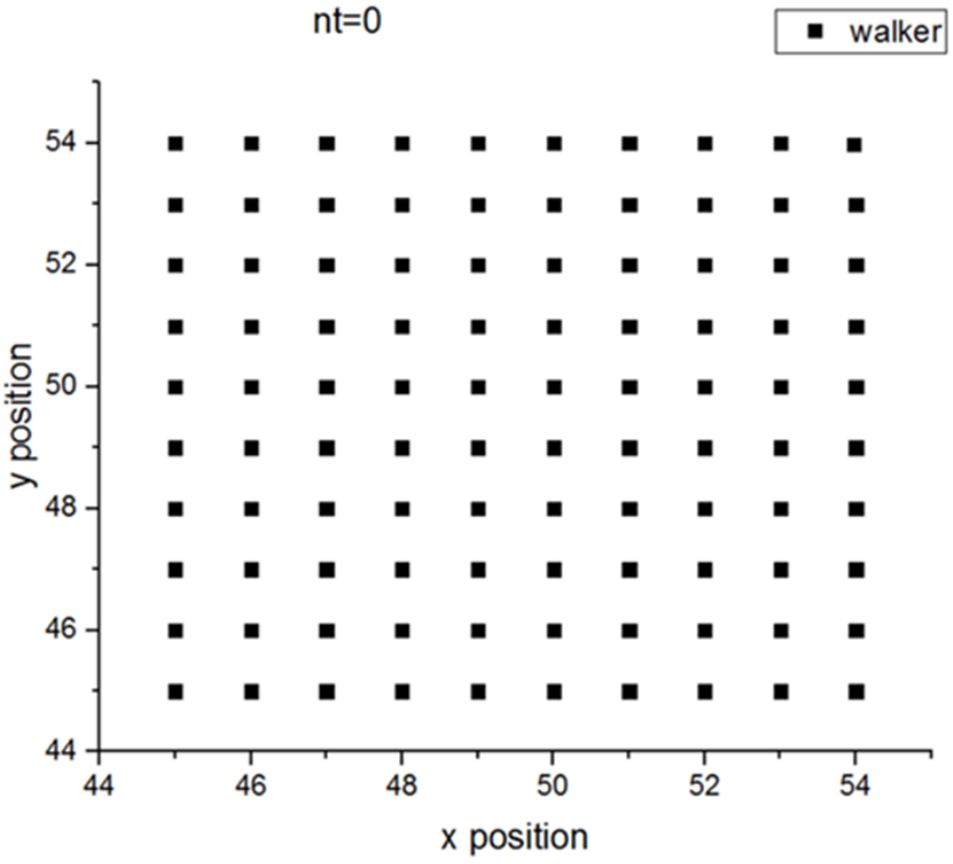

Figure 5. **Simulation for time-dependent random walk case. Number of time steps is $nt = 10$.**

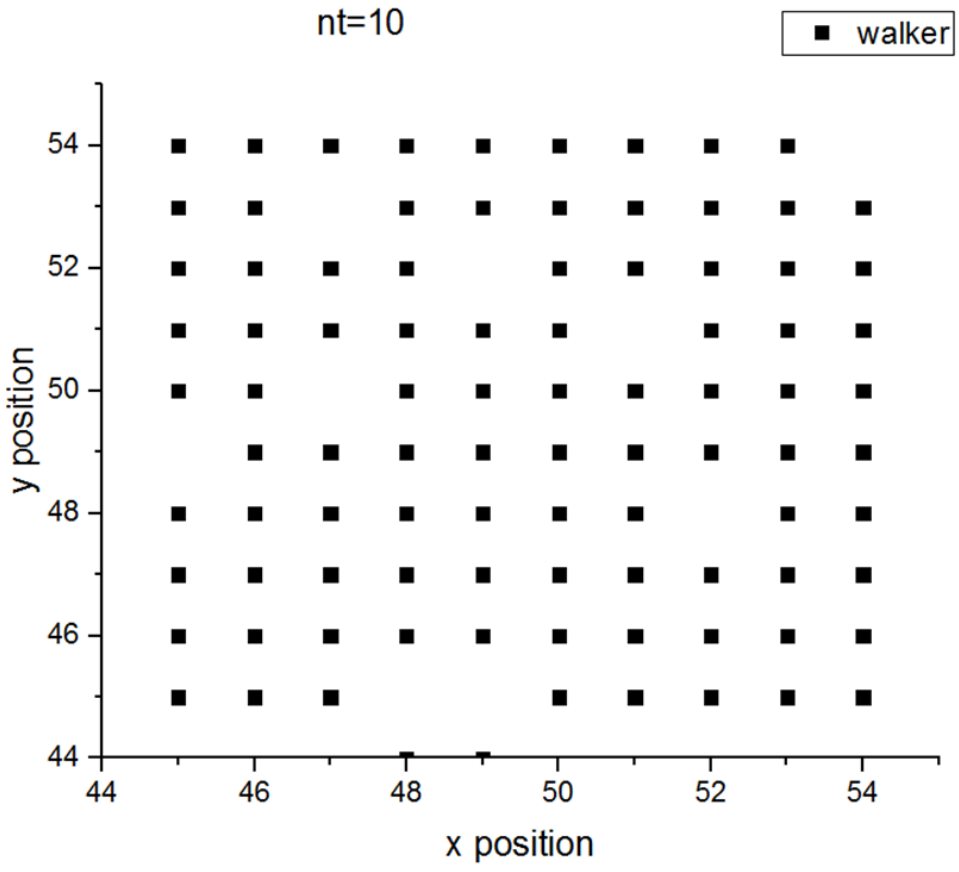

Figure 6. **Simulation for time-dependent random walk case. Number of time steps is** $nt = 100$.

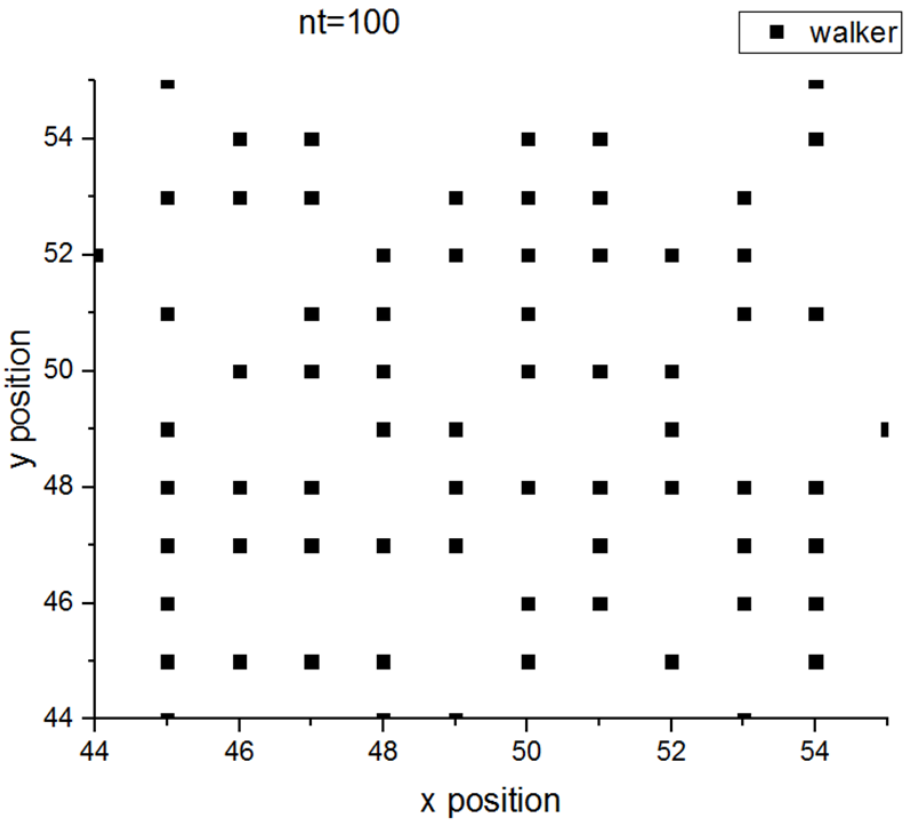

Figure 7. **Simulation for time-dependent random walk case. Number of time steps is $nt = 1,000$.**

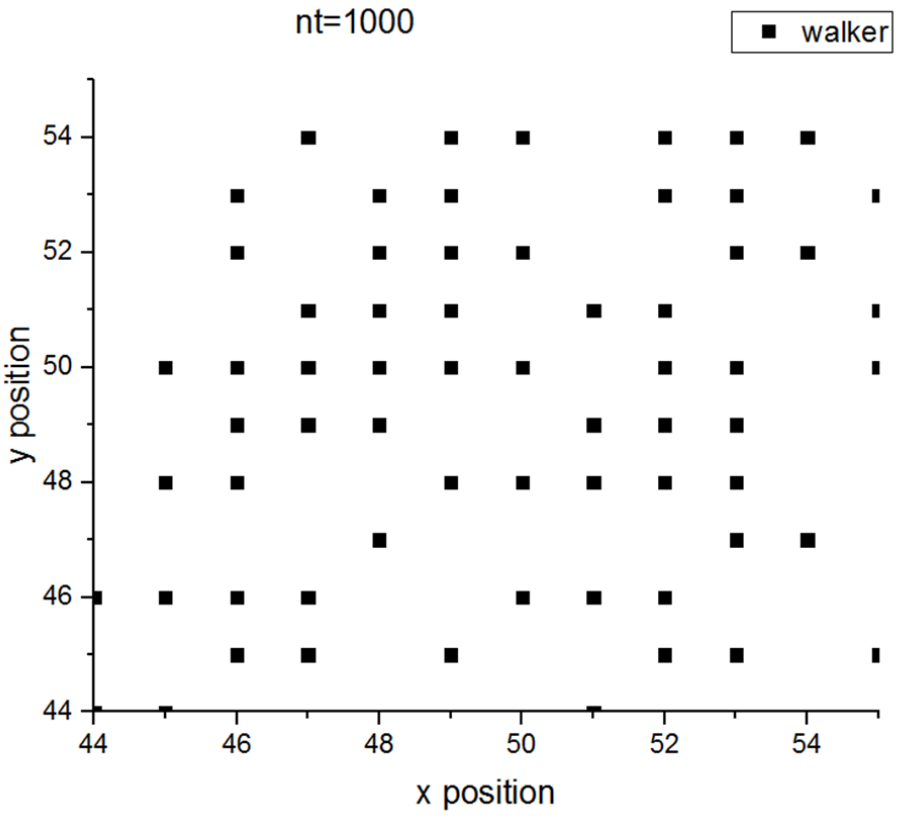

Figure 8. Simulation for time-dependent random walk case. Number of time steps is $nt = 10,000$. **Note**: Rescaled from previous plot.

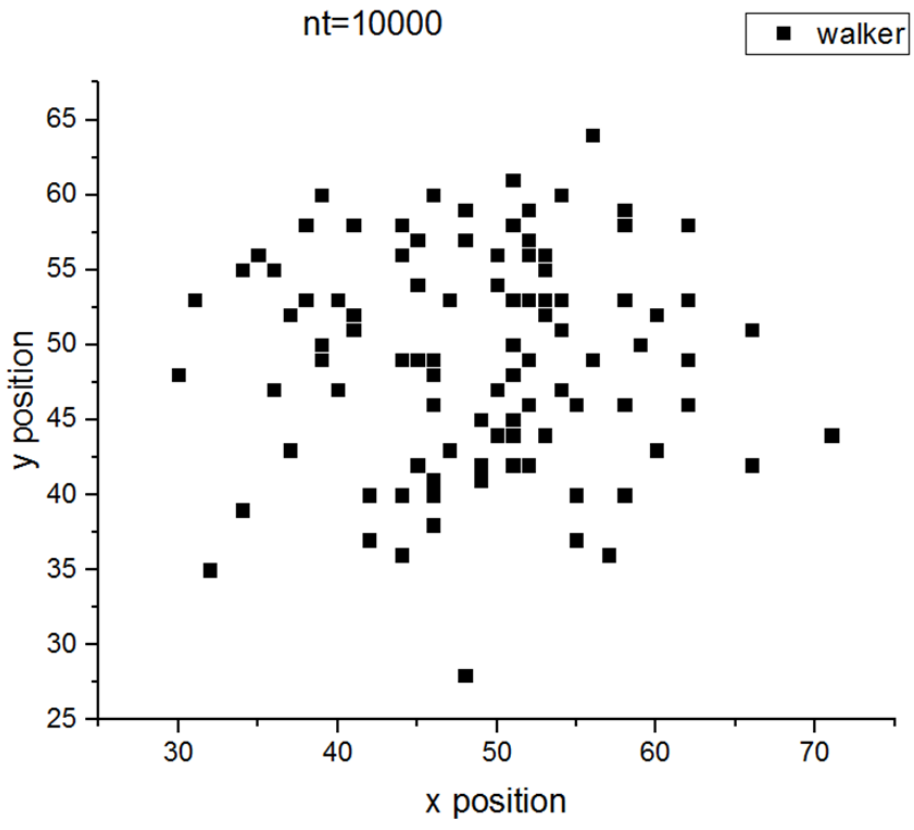

Figure 9. Simulation for time-dependent random walk case. Number of time steps is $nt = 50,000$. **Note**: Rescaled from previous plot.

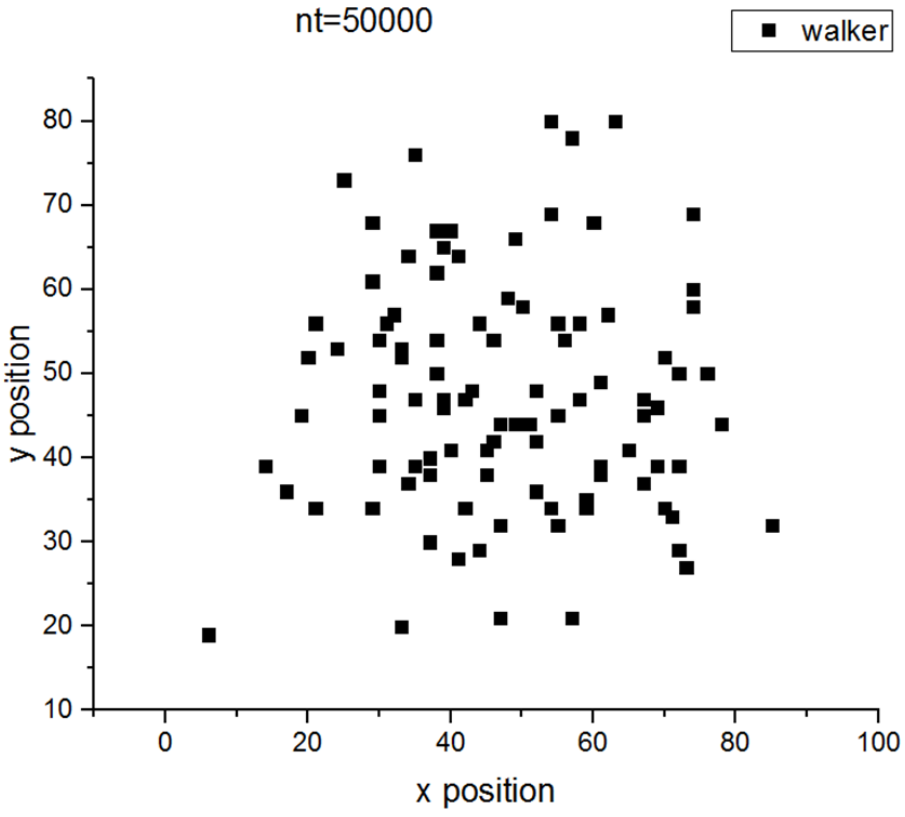

Figure 10. Simulation for time-dependent random walk case. Number of time steps is $nt = 100,000$. **Note**: Rescaled from previous plot.

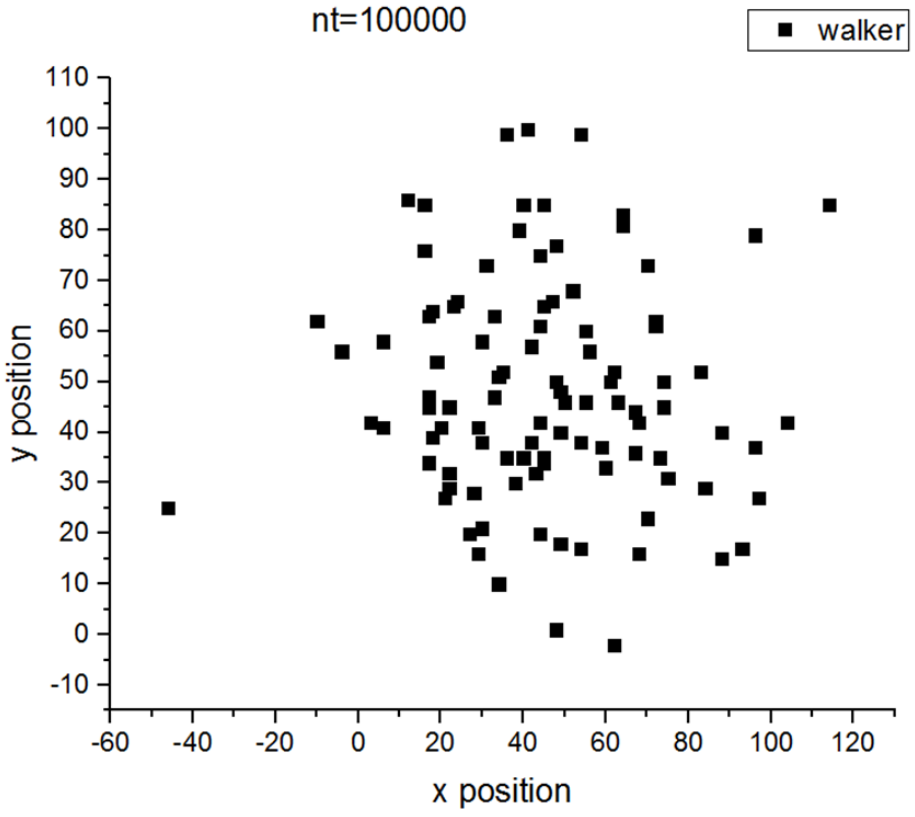

Figure 11. Simulation for $nt = 0$ case using the same scaling as the is $nt = 100,000$ case for comparison between the two. Can see how much it spreads out with time.

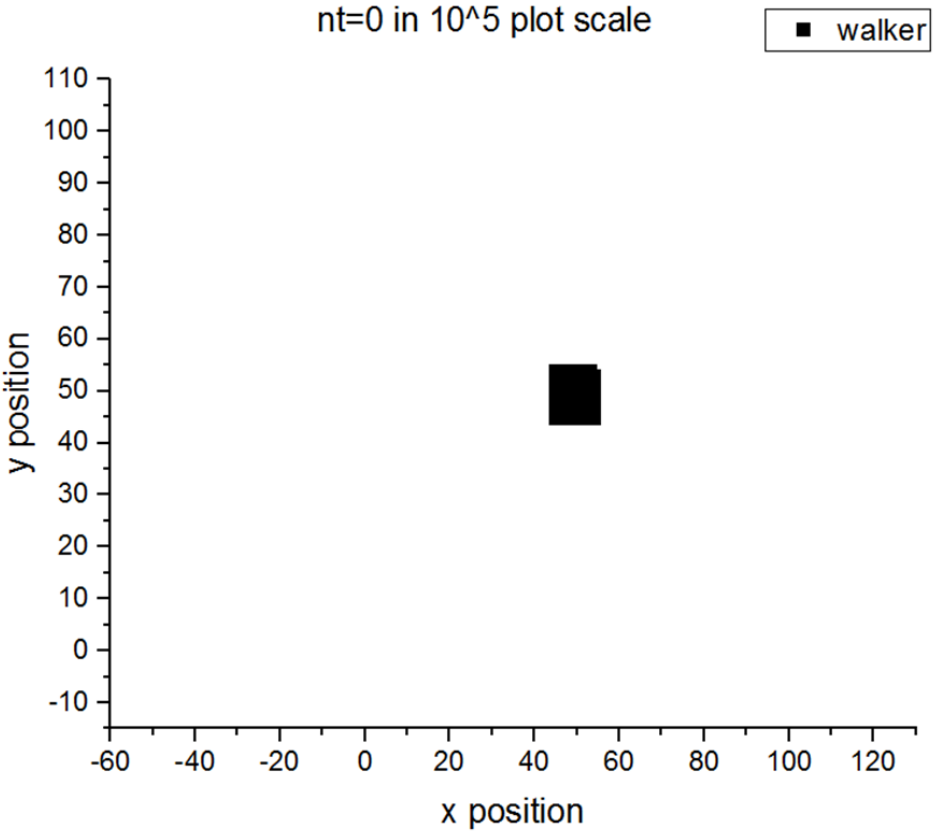

Figure 12. 1D Random walk for parameter choices $nt = 100$, $nw = 100$ = number of walkers. Notably very crowded.

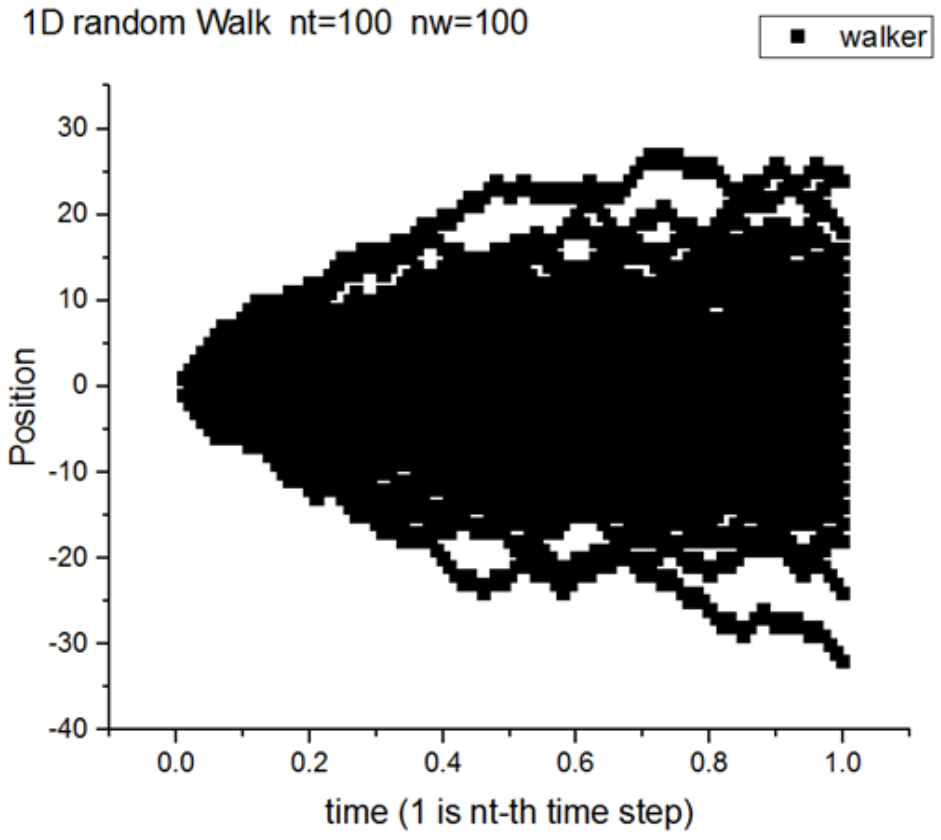

Figure 13. 1D Random walk for parameter choices $nt = 1000$, $nw = 100$ = number of walkers. Rescaled from previous 1D walk. Notably very crowded.

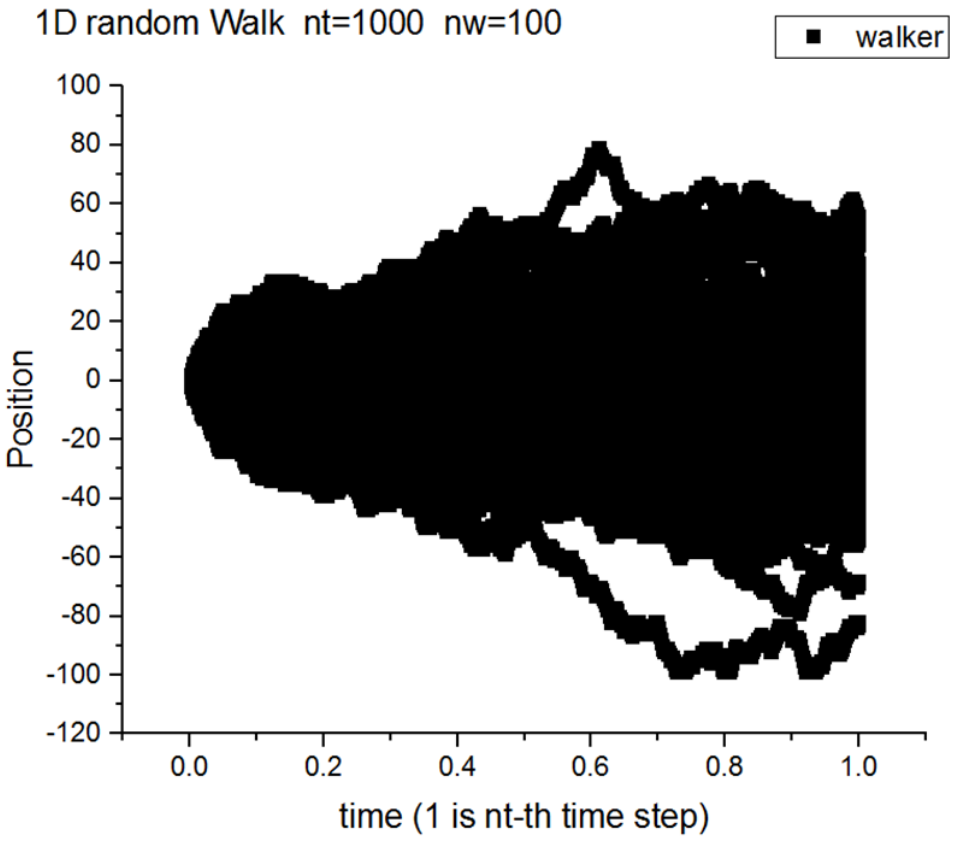

Figure 14. Mean squared deviation for 1D Random walk of parameter choices $nt = 100$, $nw = 100$ = number of walkers. Notably somewhat linear.

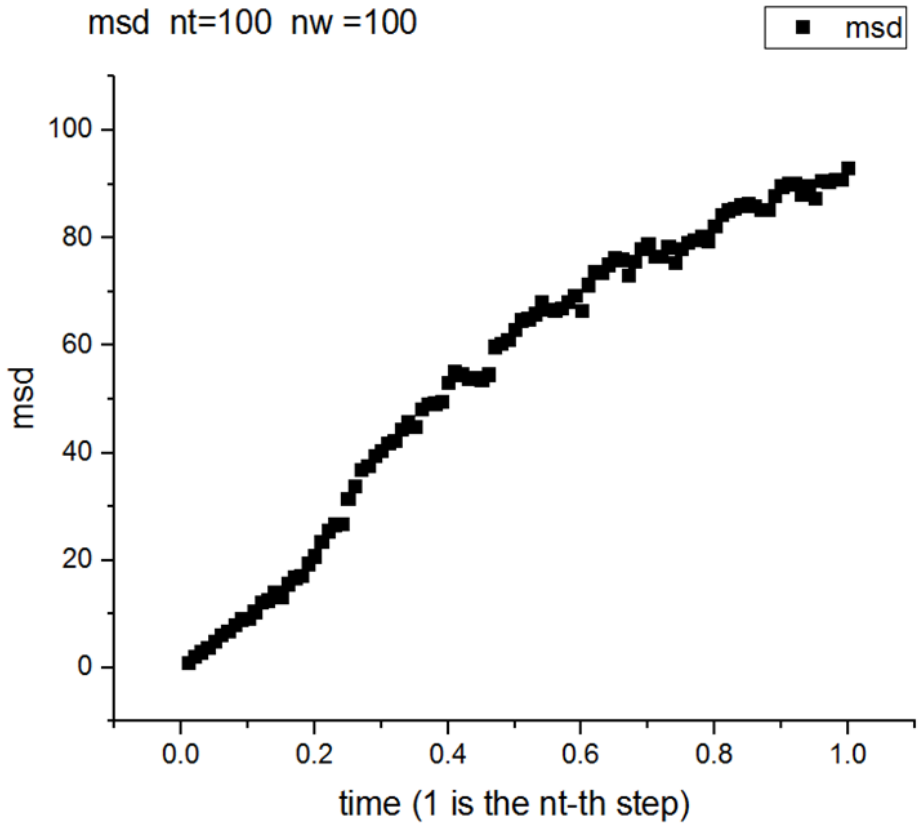

Figure 15. Mean squared deviation for 1D Random walk of parameter choices $nt = 1000$, $nw = 100$ = number of walkers. Rescaled from previous MSD. Notably somewhat linear.

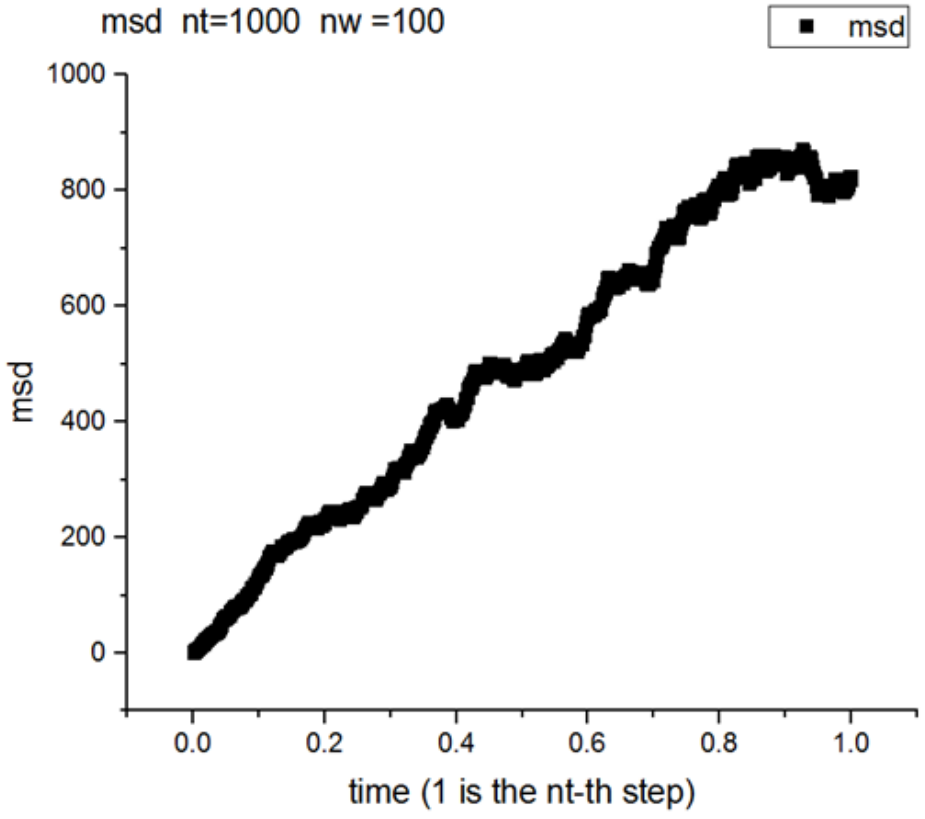

Figure 16. Code for the generation of the random 1d walks and the mean square deviation of such walks.

student12@teaching:~/proj6

```
      program randwalk1d
      real x2a(1000) ! the mean squared deviation
      print*, "Enter the number of time steps"
      read(5,*) nt
      print*, "Enter the number of walkers"
      read(5,*) nw
      open(7,file="walk_excess")
      open(8,file="msd")
  110 format(2x,e12.5,2x,i3)
  111 format(2(2x,e12.5))
      dt=1.0/nt
      x2a=0.0
      do j=1,nw ! walkers
        iy=0
        do i=1,nt ! time steps
          t=i*dt
          r1=rand(0) ! random number generation
          if(r1.le.0.5) then
            ist= 1 ! right
          else
            ist= -1 ! left
          endif
          iy= iy + ist
          x2a(i)= x2a(i) + iy*iy
          write(7,110) t,iy
        enddo
      enddo ! walkers
      do i=1,nt
        snw=nw
        x2a(i)=x2a(i)/snw
        t=i*dt
        write(8,111) t,x2a(i) ! mean sq dev w.r.t. time
      enddo
      end
```