

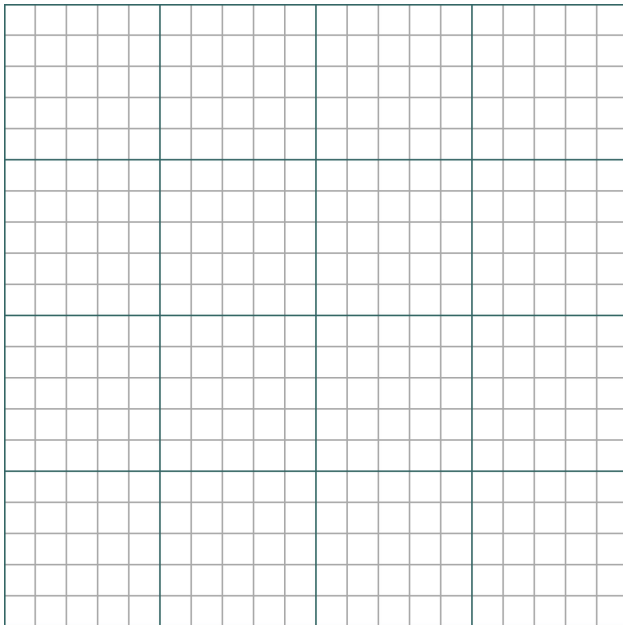
Random Walks

Random Walk Simulation Algorithm

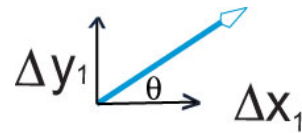
Consider again Figure 1 left. Our random walk simulation is rather simple:

- i. Start with a walker at the origin, $(x,y) = (0,0)$.
- ii. Choose a random direction $0 \leq \theta \leq 2\pi$ for the first step of the walk.
- iii. Move the walker one step in that random direction.
- iv. Increase the values for ΔX and ΔY by the values $(\Delta x = \cos \theta, \Delta y = \sin \theta)$ for that step.
- v. Repeat steps ii-iv for N steps, with the cumulative values for ΔX and ΔY being updated for each step.

Hand Exercise: Random Walk Simulation by Hand



Here are five random angles in radian measure between 0 and 2π : 2.74, 1.59, 4.70, 4.04, 0.08. Take a piece of graph paper, or print out an electronic one, and use these numbers to draw five steps in a 2-D random walk starting at the origin.

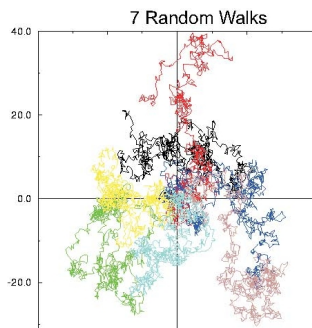


Measure the value of R for this walk and compare to Equation (15). A realistic walk might take thousands or millions of steps; be glad you do not have to draw that by hand.

Random Walk Simulation Algorithm

Part I, Go Take a Walk

1. Locate the random-number generator in whatever software system you are using.



2. If possible, determine how to use a different seed for each random walk. If you start your simulation from scratch each time, it will give the same results each time, and the averaging will not work properly. If you just continue using the generator already in

use, then the sequence of random numbers probably will not repeat.

3. Since each step is of length 1, for step i , $\Delta x_i = \cos \theta_i$ and $\Delta y_i = \sin \theta_i$.
4. Sometimes you can tell your random number generator the range of values which you wish your random number r_i to span. If so choose the range $0 \leq r_i \leq 1$. In that case a random angle that covers the entire 2-D plane is just:

$$(16) \quad \theta_i = 2\pi r_i$$

5. Plot a line from the end of the previous step to the end of the new step, like in Figure 3. (Usually this is done just by plotting the points and having the plotting program connect the dots.)
6. Update your values of ΔX and ΔY , the total distances covered in the x and y directions, to include the latest step.
7. Continue the walk and the plot until your walker has made N steps ($N = 200$ is a reasonable value to start with, although larger values are better).
8. When your walker has completed N steps, calculate the value of R^2 according to Equation (6). Record (preferably on the computer) the values of N , R^2 , ΔX and ΔY . Note that ΔX and ΔY are the total lengths in the x and y directions, and can be positive or negative.
9. Start your walker again at the origin. Using a different sequence of random numbers than in the previous walk. Have your walker complete another walk of N steps, and again record the values of N , R^2 , ΔX and ΔY when the walk is over.
10. With the same value of N that you have been using, complete a total of at least 100 different random walks (more if your walk takes more than 200 steps).
11. Calculate and record the mean value of R^2 for all the walks according to Equations (8) and (9).
12. Next calculate the average values of ΔX and ΔY for these 100 walks.
13. If your walks are random, we would expect that $\overline{\Delta X} \approx \overline{\Delta Y} \approx 0$. You should obtain a "small" value for these quantities, if not exactly zero. To be more scientific about what we mean by "small", calculate $\Delta X/\sqrt{R^2}$ and $\Delta Y/\sqrt{R^2}$. These are now dimensionless and measured in the proper relative scale to be compared to 1. For 100 walks we would expect small to be ≈ 0.1 ; for 10,000 walks we would expect small to be ≈ 0.01 . (This is based on the statistical result that statistical deviations of relative values behave like $1/\sqrt{N}$.)
14. Use the values of R^2 obtained for each walk to obtain a value for the rms radius for this N value according to Equation (9), $R_{rms}(N) = \sqrt{R^2(N)}$.

Part II, Does the Simulation Agree with the Math?

The mathematics predicts that if the walks are random, then the root-mean square distance from the origin, averaged over many walks, should approximately agree with Equations (15) and (16)

$$(15) \quad R_{rms}(N) = \sqrt{\overline{R^2}(N)} \approx \sqrt{N}$$

$$(16) \quad \overline{\Delta X} \approx \overline{\Delta Y} \approx 0 .$$

Equation (16) says that the average length of the vector from the origin vanishes, $\overline{R}(N) \approx 0$, since positive and negative values for the components are equally likely. However, equation (15) tells us that the average distance from the origin does not vanish. The verification that your simulation agree with theses relations, at least approximately and within statistical errors, would give us confidence that the pseudorandom numbers we are generating are a good representation of truly random numbers, and that our random walk does indeed simulate those walks occurring in nature.

The key concept to understand in the test of (15) is that *there are two separate and independent parts* involved.

1. First, for a fixed number of steps N , you need to average R^2 over many walks in order to obtain and average value $\overline{R^2}(N)$ as a function of the number of steps.
2. Second, you need to vary the number of steps N and calculate $\overline{R^2}(N)$ for repeated walks at this new value of N . These walks must be different from those in part 1 in order to maximize randomness.

After you have completed all the averages, you take the square root of the mean squared value $R^2(N)$ for each N and see how well Equation (15) is satisfied.

3. Make a plot of $R_{rms}(N)$ versus \sqrt{N} . If your results scatter about a straight line, then you have verified that your simulation agrees with the mathematical model, and presumably nature.