Lecture 7 Random systems

Random Numbers

Random number generator in Mathematica: RandomReal[]. Random number generated is uniformly distributed in [0,1].

We could also generated random number distributed as a Gaussian between a given interval, using the command `NormalDistribution`. See sample code 7.1 – 7.3.1.

Stochastic system.

In this chapter we consider a class of systems in which randomness plays a central role. These are called random or *stochastic* systems. Typically, these systems consist of a very large number of "degrees of freedom," which might be associated with particles (e.g., in a macroscopic sample of a liquid) or perhaps spins (in a piece of a ferromagnet). Randomness can then arise in several different ways. For example, it might be impossible to obtain complete knowledge of the positions and velocities of all of the particles. Or, our system may interact with a thermal reservoir in the complicated manner that is best described using probabilities and averages as familiar from statistical mechanics.[2] Hence, even though the "underlying" physics of the system may be deterministic, our incomplete knowledge may force us to resort to a statistical, i.e., stochastic, description. Fortunately, as we will see, a statistical description is often extremely useful, and indeed, most appropriate in these problems.

Our point is that such a complete computational solution of the problem would give far more information than we want or need to understand the mixing process. What we really require is a *statistical* description, or theory, of the behavior. We don't care about the detailed trajectory of every cream particle. It is enough for us to know the average properties of a particle trajectory. Since the number of particles involved is very large, these averages will be very sharply defined, and if we are careful we can also estimate the fluctuations from the averages using statistical arguments.

A typical stochastic problem is diffusion; this describes such important processes as the spreading of a drop of cream in your morning coffee.

Each cream molecule follows a complicated trajectory as it collides repeatedly with other molecules. Such a trajectory can be described by what is known as a "random walk." This is a process in which a particle (the walker) moves one step at a time, according to certain rules. In the model appropriate for our cream molecule, the walker's steps correspond to the motion of the molecule between collisions. Each collision changes the direction of the velocity of the molecule, and this is modeled by letting the direction of each step in the walk be random. Hence, our random walker follows a zig-zag path that is similar to the effectively random trajectory of a cream molecule.

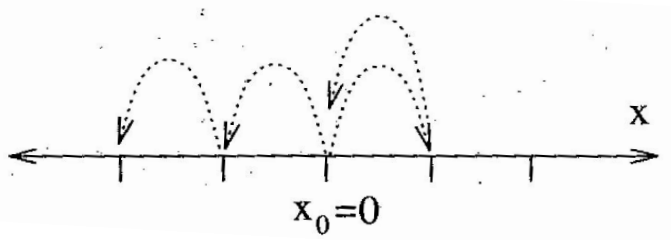


FIGURE 7.1: Sketch of a random walk in one dimension. The walker began at $x = x_0 = 0$, and each step is indicated schematically by a dotted arrow. Here the first step happened to be to the right, while the next three steps were to the left.

EXAMPLE 7.1 Routine `rwalk` for simulating one-dimensional random walks and calculating the mean-squared displacement

- Initialize a random number generator (say, rnd) (if necessary).
- Initialize an array $x2ave(i) \equiv 0$ $(i = 1, 2, ..., n)$ which will hold the squared displacements at time step i.
- Loop through the desired number of walkers $(j = 1, ..., m)$.
  - ▷ Initialize the initial location of the walker to $x = 0$. (Note that you need not keep the locations of all of the walkers at all steps. You can simply accumulate the squared displacement at step i into the array $x2ave(i)$ as soon as it becomes available and update x for the $i + 1$-st step, etc.)
  - ▷ Loop through the number of steps ($i = 1$ through n) in each walk.
    - Get a random number $r = rnd$ between 0 and 1.
    - If $r < 0.5$, update x to $x + 1$, otherwise to $x - 1$.
    - Accumulate the squared displacement: $x2ave(i) = x2ave(i) + x^2$.
- Normalize the squared displacements to get the *mean* squared displacement averaged over all of the walkers: $x2ave(i) = x2ave(i)/m$ for all $i = 1, ..., n$.

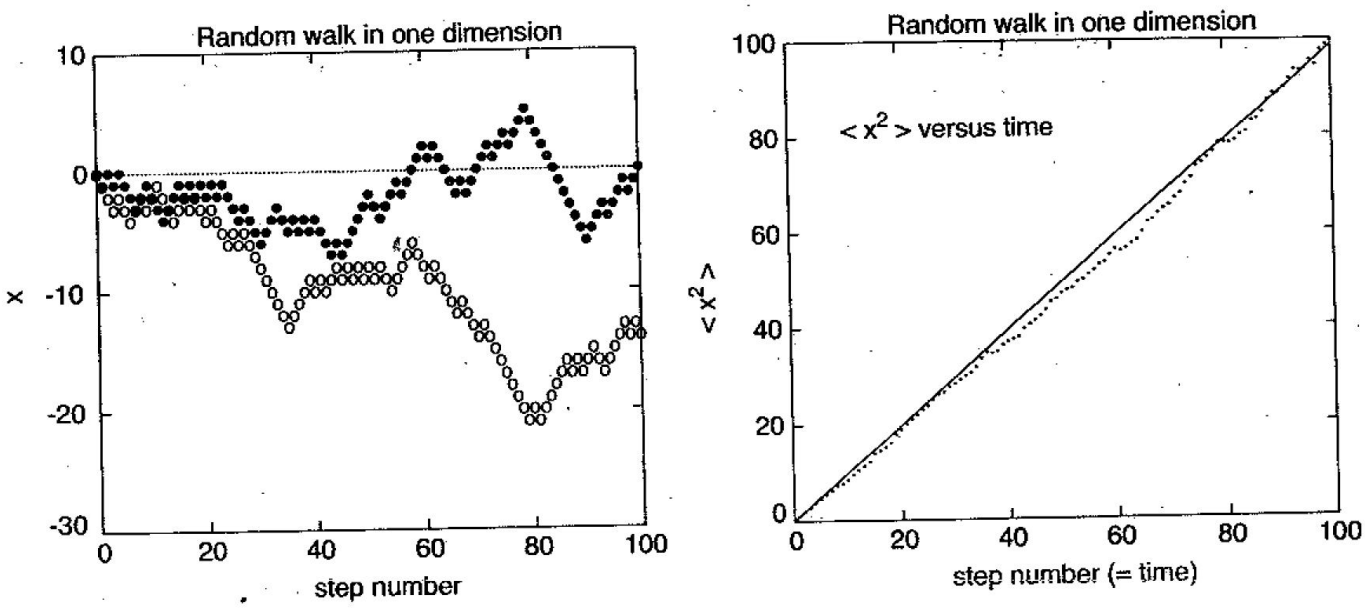


FIGURE 7.2: Left: x versus step number, that is, time, for two random walks in one dimension. Right: $\langle x^2 \rangle$ as a function of step number (which is proportional to time) for a collection of one-dimensional random walks. The step length was unity and the results for 500 walkers were averaged. The points are the calculated values and the straight line is a least-squares fit to the form (7.1).

the most basic is the mean displacement of a walker after n steps. Since a walker is as likely to step left as right, this average, which we denote by $\langle x_n \rangle$, must be zero.[4]

The random walk code of a single walker can be found in sample code 7.4.

Fluctuation in *x*

Consider the displacements of jlast walkers at any instance, x[j]_n.

$\langle x \rangle_n = \dfrac{\sum_{j=1}^{jlast} x_n[j]}{jlast} = 0$ since the walker has equal probability to walk left and right.

We can also calculate $\langle x^2 \rangle_n = \dfrac{\sum_{j=1}^{jlast} x_n^2[j]}{jlast}$.

The variance in x is defined as $\sigma_x^2 = \langle x^2 \rangle_n - (\langle x \rangle_n)^2$. Note that $\sigma_x^2 = \sigma_x^2(t)$, and is evaluated at fixed time by summing over all walkers' contribution. The variance in x, σ_x^2, is sometimes referred to 'fluctuation of x'.

$\sigma_x = \sqrt{\langle x^2 \rangle_n - (\langle x \rangle_n)^2}$ is known as the root mean square of the displacement.

Diffusion

We can show that the variance at time n is given by $\langle x^2\rangle - \langle x\rangle^2 = 2Dt$
D = diffusion constant. Here, n and t are used interchangeably. The value of D can be calculated analytically. Writing the position after n steps, x_n, as a sum of n separate steps gives

$$x_n = \sum_{i=1}^{n} s_i \,, \tag{7.8}$$

where s_i is displacement for the ith step. For this problem $s_i = \pm 1$ with equal probabilities. We can then write

$$x_n^2 = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} s_i \, s_j \right) . \tag{7.9}$$

Since the steps are independent of each other, the terms $s_i s_j$ with $i \neq j$ will be ± 1 with equal probability. If we average x^2 over a large number of separate walks this will leave only the terms s_i^2. Thus we find

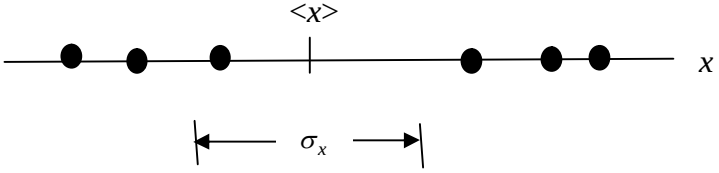
$$< x_n^2 > = \sum_{i=1}^{n} s_i^2 = n \,, \tag{7.10}$$

where we have used the fact that $s_i^2 = 1$. Since n is also equal to time, this is identical to with $\langle x^2\rangle - \langle x\rangle^2 = 2Dt$, with D = ½. Note that we have made use of the fact that $\langle x\rangle_n = \dfrac{\sum_{j=1}^{jlast} x_n[j]}{jlast} = 0$.

$\sigma_x = \sqrt{\langle x^2\rangle_n - \left(\langle x\rangle_n\right)^2}$

should be contrasted to the displacement of a free particle with initial velocity v_0. In that case the distance from the initial position after a time t is $x(t) = vt$ whereas for a diffusion process the root mean square value is $\sqrt{\langle x^2\rangle - \langle x\rangle^2} \propto \sqrt{t}$. Since diffusion is strongly linked with random walks, we could say that a random walker escapes much more slowly from the starting point than would a free particle. We can vizualize the above in the following figure. In

The root mean square displacement after a time t $\sigma_x = \sqrt{\langle x^2\rangle_n - \left(\langle x\rangle_n\right)^2} = \sqrt{2Dn}$, can be interpreted as the deviation of the displacement of any single walker from the sample's averaged displacement $\langle x\rangle_n$ at time n. σ_x measures how far a typical walker is 'drifted' away from its original position.



It is found that the fluctuation in *x* grows with *n*, $\sigma_x^2 \propto t$. This also implies that the separation of two typical walkers will grows as *n* increases.

All in all, the random walkers behave like what we see in diffusion behaviour of particles such as that found in a drop of cream in a cup of coffee. The behavior of diffusion is characterized by $\langle x^2 \rangle_n = 2Dn$.

See sample code 7.5 for the simulation of a collection of jlast walkers. The averaged displacement-squared $\langle x^2 \rangle_n$ as a function of time is simulated.

2D Random walk.

The random walk in 1-D can be easily generalized to 2-D.

Exercise 10.1

Extend the above program to a two-dimensional random walk with probability $1/4$ for a move to the right, left, up or down. Compute the variance for both the x and y directions and the total variance.

(Jensen, page 172)

See sample code 7.6.

Diffusion

See chpater7a.nb

Random walk and diffusion

See chpater7a.nb