3. Random processes

What is noise?

- We discussed chaos vs noise in the previous lecture
- The difference between chaos and noise?
- The use of noise in simulations in physics

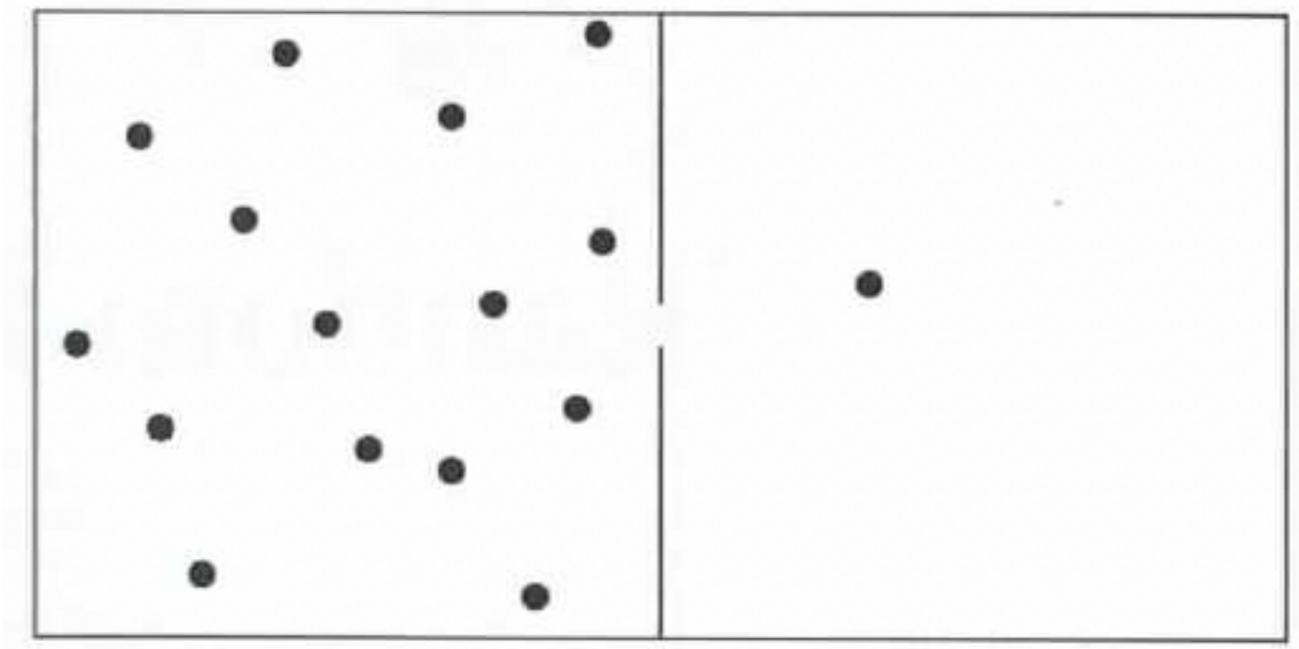
Random processes

- Truly random: Nuclear decay
- Inaccuracies: Throwing of darts
- Sum of many deterministic processes: diffusion of colloidal particles

Ways to model random processes

- Monte Carlo: equilibrium distributions or quasi-dynamics
- Brownian dynamics: dynamic, no intertia
- Langevin dynamics: dynamic, interia

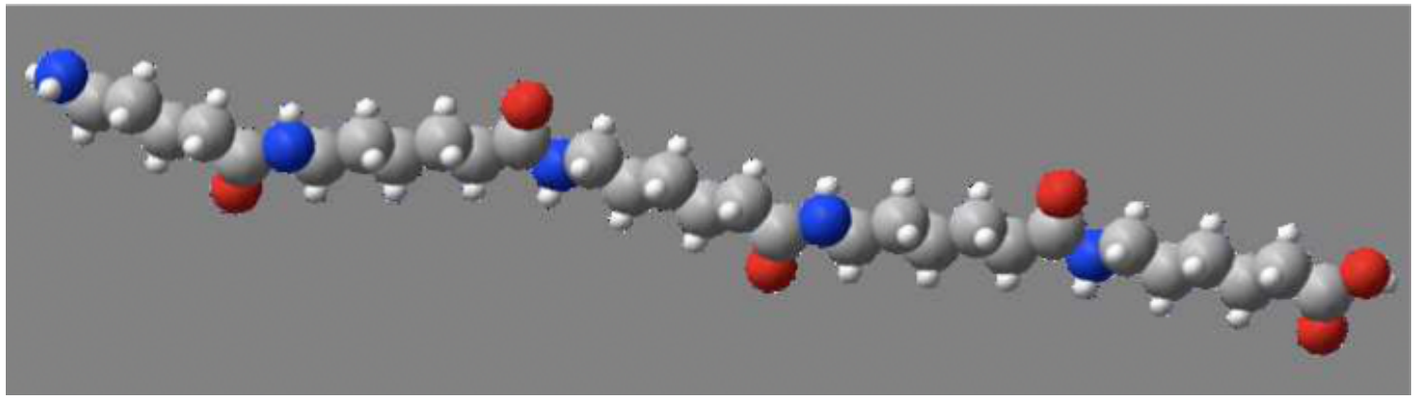
Particles in a box



- The passing of a single particle is a random event
- But passing of many particles over time is less random
- The collective effects of many random events can produce precise and accurately measurable quantities

Polymers

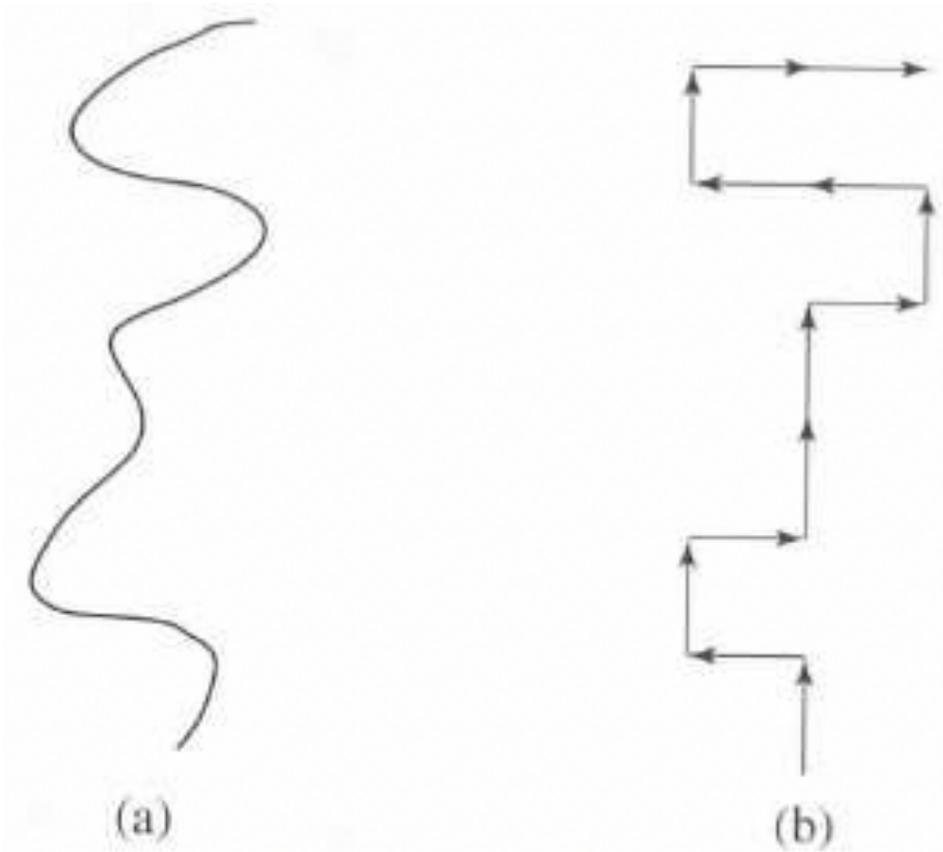
- Example: nylon



- Polyelectrolytes, applications e.g.

Polymer chain in good solvent

- "Good" solvent means that the polymers "feels" no difference in interaction between itself and the solvent
- You can approximate a polymer by a discrete random walk



Self-avoiding random walk

- In dense polymer melts it is important the parts of the chain don not occupy the same space ⇒ use

  a self-avoiding random walk

- Important: when a random walk revisits a point, you have

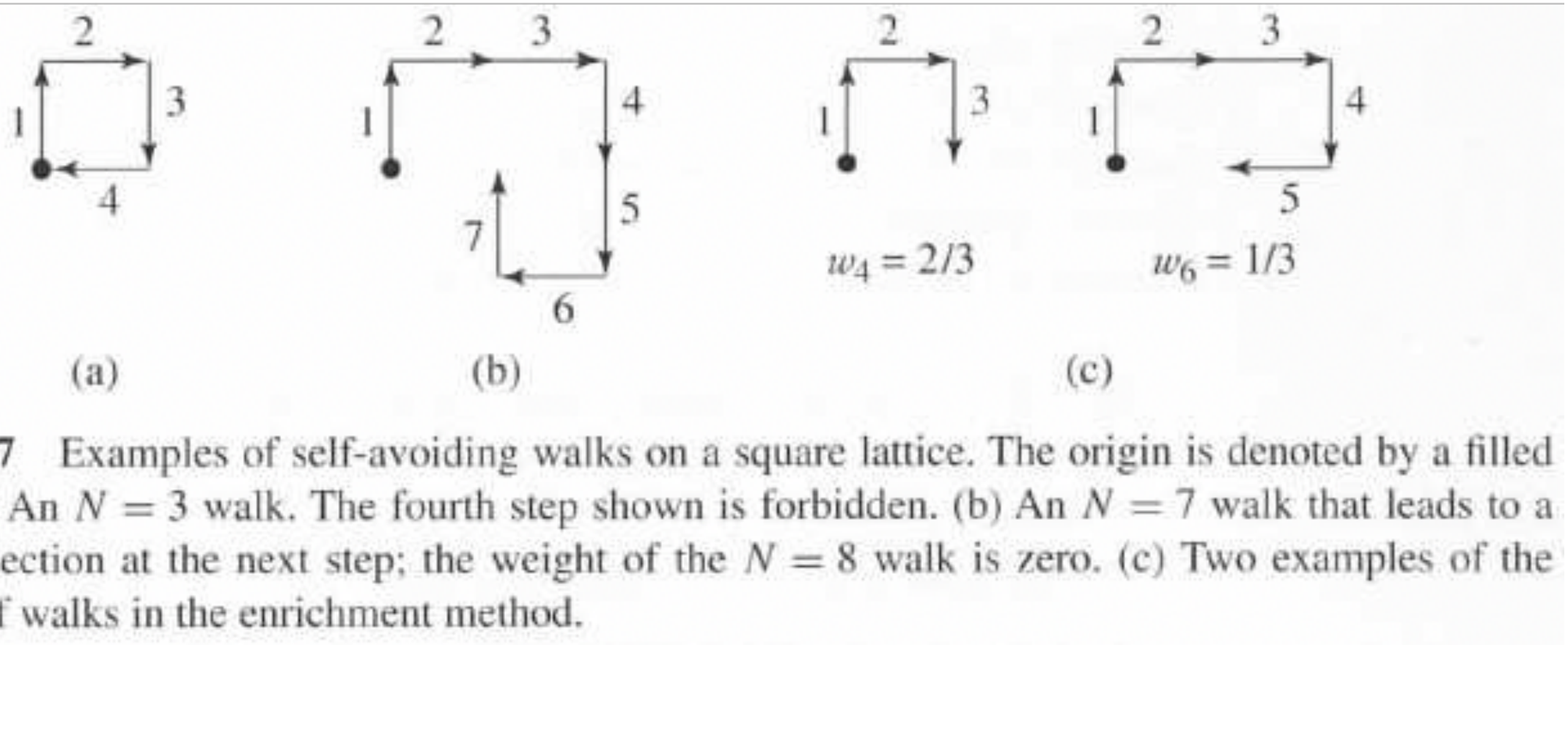


Figure 7.7 Examples of self-avoiding walks on a square lattice. The origin is denoted by a filled circle. (a) An $N = 3$ walk. The fourth step shown is forbidden. (b) An $N = 7$ walk that leads to a self-intersection at the next step; the weight of the $N = 8$ walk is zero. (c) Two examples of the weights of walks in the enrichment method.

Reptation model

- Example: self-avoiding random walk with only 90° bends
  A reptation step (can be alternated with growing steps):
  - take the tail link and move it to the head at ±90°
  - if it self-intersects, take the original chain and interchange head and tail

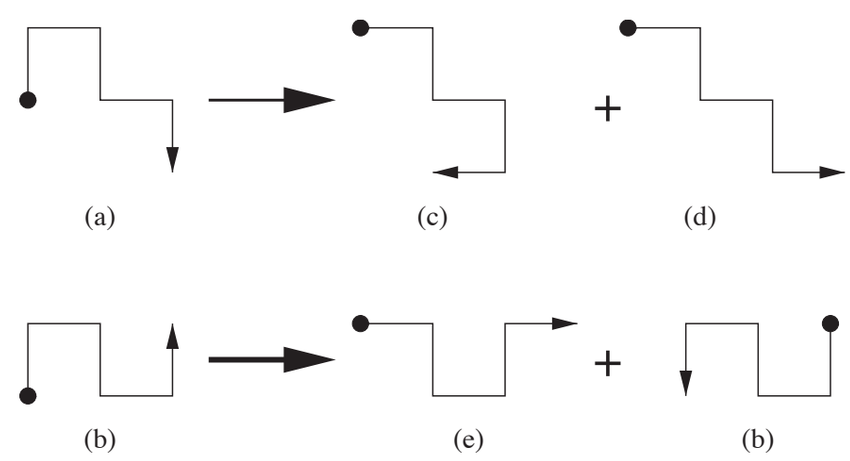


Figure 7.9: The possible transformations of chains a and b. One of the two possible transformations of chain b violates the self-intersection restriction and the head and tail are interchanged.

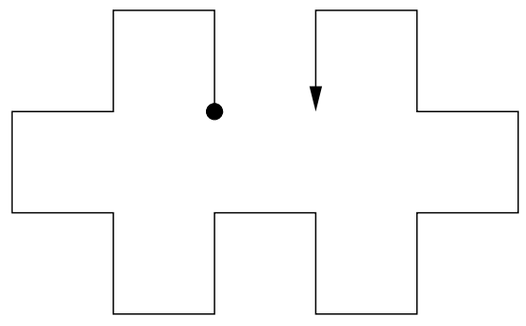


Figure 7.10: Example of a double cul-de-sac configuration for the self-avoiding walk that cannot be obtained by the reptation method.

Pseudo random number generators

- Random number generators on computers are usually deterministic and therefore not really random. But we call them simply random anyhow, since computers almost never had real random number generators.
- Some possibilities for real random generators:
  - measure electric or temperature fluctuations
    - present in most modern processors

Statistical tests of randomness

a. *Period.* An Obvious requirement of a random number generator is that its period be much greater than the number of random numbers needed in a specific calculation.

Statistical tests of randomness

b. *Uniformity.* A random number sequence should contain numbers distributed in the unit interval with equal probability. The simplest test of uniformity is to divide this interval into M equal size subintervals or bins. For example, consider the first $N = 10^4$ numbers generated by (7.58) with $a = 106$, $c = 1283$, and $m = 6075$ (see Press et al.). Place each number into one of $M = 100$ bins. Is the number of entries in each bin approximately equal? What happens if you increase N?

c. *Chi-square test.* Is the distribution of numbers in the bins of part (b) consistent with the laws of statistics? The most common test of this consistency is the *chi-square* or χ^2 test. Let y_i be the observed number in bin i and E_i be the expected value. The chi-square statistic is

$$\chi^2 = \sum_{i=1}^{M} \frac{(y_i - E_i)^2}{E_i}. \tag{7.60}$$

For the example in part (b) with $N = 10^4$ and $M = 100$, we have $E_i = 100$. The magnitude of the number χ^2 is a measure of the agreement between the observed and expected distributions; chi^2 should not be too big or too small. In general, the individual terms in the sum (7.60) are expected to be order one, and because there are M terms in the sum, we expect $\chi^2 \leq M$. As an example, we did five independent runs of a random number generator with $N = 10^4$ and $M = 100$, and found $\chi^2 \approx 92$, 124, 85, 91, and 99. These values of χ^2 are consistent with this expectation. Although we usually want χ^2 to be as small as possible, we would be suspicious if $\chi^2 \approx 0$, because such a small value suggests that N is a multiple of the period of the generator and that each value in the sequence appears an equal number of times.

d. *Filling sites.* Although a random number sequence might be distributed in the unit interval with equal probability, the consecutive numbers might be correlated in some way. One test of this correlation is to fill a square lattice of L^2 sites at random. Consider an array $n(x, y)$ that is initially empty, where $1 < x_i, y_i < L$. A site is selected randomly by choosing its two

Statistical tests of randomness

e. *Parking lot test.* Fill sites as in part (d) and draw the sites that have been filled. Do the filled sites look random, or are there stripes of filled sites? Try $a = 65549$, $c = 0$, and $m = 231$.

f. *Hidden correlations.* Another way of checking for correlations is to plot x_{i+k} versus x_i. If there are any obvious patterns in the plot, then there is something wrong with the generator. Use the generator (7.58) with $a = 16807, c = 0$, and $m = 2^{31} - 1$. Can you detect any structure in the plotted points for $k = 1$ to $k = 5$? Test the random number generator that you have been using. Do you see any evidence of lattice structure, for example, equidistant parallel lines? Is the logistic map $x_{n+1} = 4x_n(1 - x_n)$ a suitable random number generator?

g. *Short-term correlations.* Another measure of short term correlations is the autocorrelation function

$$C(k) = \frac{\langle x_{i+k}x_i\rangle - \langle x_i\rangle^2}{\langle x_i x_i\rangle - \langle x_i\rangle\langle x_i\rangle}, \tag{7.61}$$

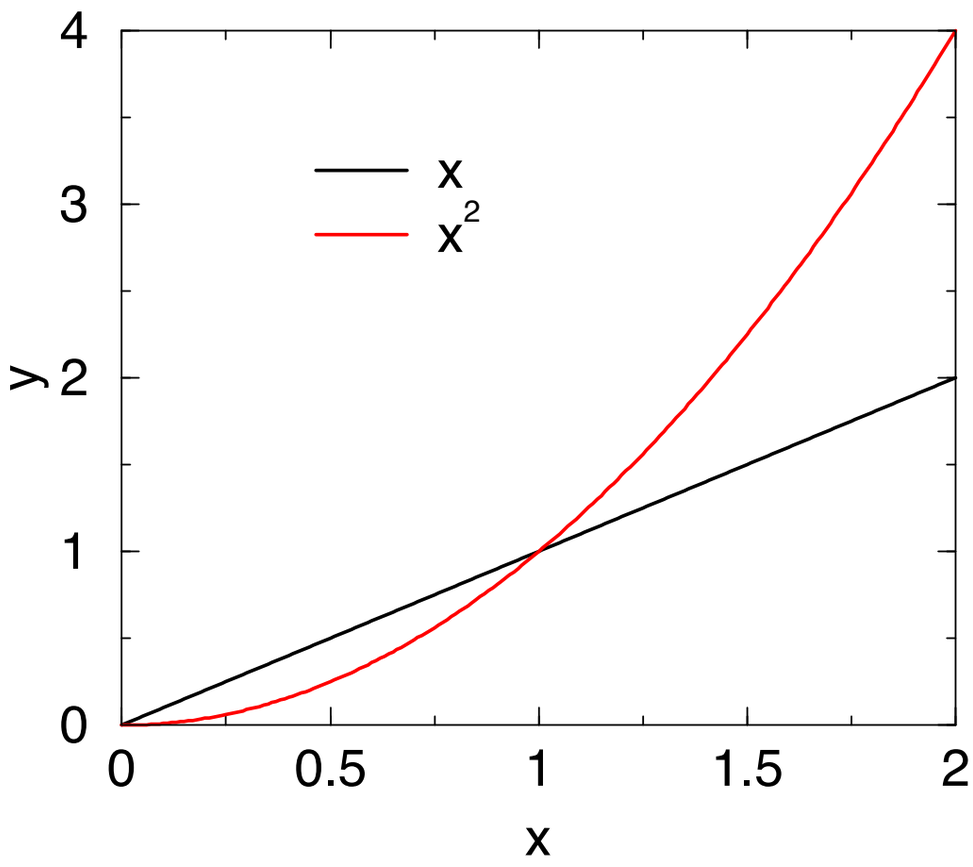
where x_i is the ith term in the sequence. We have used the fact that $\langle x_{i+k}\rangle = \langle x_i\rangle$, that is, the choice of the origin of the sequence is irrelevant. The quantity $\langle x_{i+k}x_i\rangle$ is found for a particular choice of k by forming all the possible products of $x_{i+k}x_i$ and dividing by the number of products. If x_{i+k} and x_i are not correlated, then $\langle x_{i+k}x_i\rangle = \langle x_{i+k}\rangle\langle x_i\rangle$ and $C(k) = 0$. Is $C(k)$ identically zero for any finite sequence? Compute $C(k)$ for $a = 106$, $c = 1283$, and $m = 6075$.

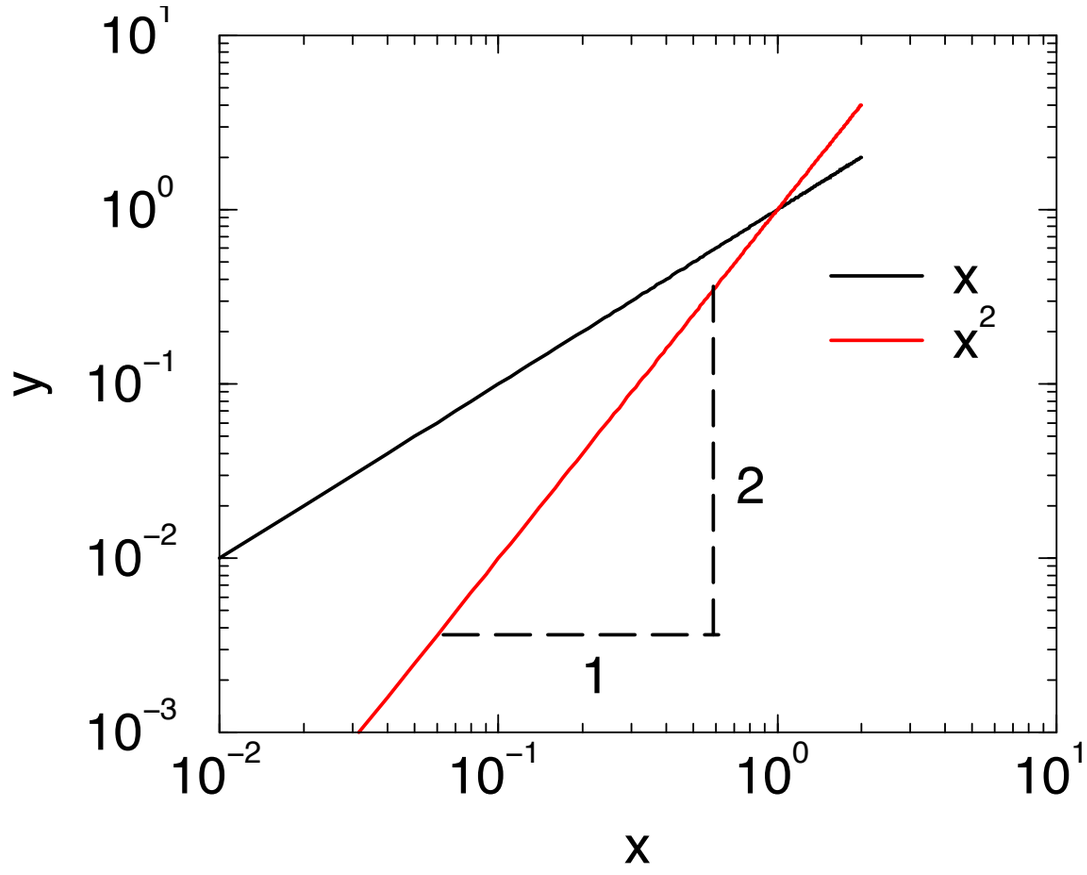
h. *Random walk.* A test based on the properties of random walks has been proposed by Vattulainen et al. Assume that a walker begins at the origin of the x-y plane and walks for N steps. Average over M walkers and count the number of walks that end in each quadrant q_i. Use the χ^2 test (7.60) with $y_i \to q_i$, $M = 4$, and $E_i = M/4$. If $\chi^2 > 7.815$ (a 5% probability if the random number generator is perfect), we say that the run fails. The random number generator fails if two out of three independent runs fail. The probability of a perfect generator failing two out of three runs is approximately $3 \times 0.95 \times (0.05)^2 \approx 0.007$. Test several random number generators.

Required “randomness”

- The requirements for a random generator depend completely on the application.
- There are some general rules for random generators and there are many good ones, but also many bad ones around.
- There can be a trade off between “accuracy” and speed.

Log-log scale





References

- Project report on several methods to generate self avoiding walks: http://www.iskp.uni-bonn.de/uploads/media/polymers.pdf
- Movie of a 100 million step 3D random walk: http://www.youtube.com/watch?v=GHzfadEuk1U

Projects: Particles in a box

3.1) Use the template and add the decision of moving a particle. Because each particle has the same chance to go through the hole, the probability that a particle goes from left to right equals the number of particles on the left divided by the total number of particles, that is $p = n/N$. Thus:

- Generate a random number r from a uniform distribution between 0 and 1
- If $r \leq p = n/N$, move a particle from left to right $n \rightarrow n - 1$, otherwise to the left $n \rightarrow n + 1$
- Run simulations for N= 8, 64 and 800

Does the system reach equilibrium? What is your qualitative criterion for equilibrium? Does n, the number of particles on the left-hand side, change when the system is in equilibrium?

3.2) Does the time dependence of n appear to be deterministic for sufficiently large N? What is the qualitative behavior of n(t)?

3.3) Now look at the effect of seeding the random generator. What does random.seed() exactly do in python (or equiv. in MatLab)?

What is the effect of the time behavior of the first simulation? What is the effect on the histogram of the equilibrium end values of many simulations?

3.4) A measure of the equilibrium fluctuations is the mean square fluctuations $\Delta n^2 = \langle (n - \langle n \rangle)^2 \rangle = \langle n^2 \rangle - \langle n \rangle^2$. Determine this when averaging over the end values of many simulations as well as for averaging over time for a long, equilibrium part of a single simulation. How do these two different fluctuations compare? How do they depend on N?

3.5) Compute the average $\langle n \rangle$ from the end values of many simulations as well as for averaging over time for a long, equilibrium part of a single simulation. Which average might be more accurate?

Projects: Polymers as 2-dimensional random walks

3.6) Write a program that generates a two-dimensional random walk with single steps along x or y. The simplest way to do this is by generating a number randomly taken out of the sequence (0,1,2,3) by multiplying random.rnd() by 4 and rounding the result down to an integer using int(). Then you can increase x by one for 0, decrease x by one for 1, and the same for y with 2 and 3. Plot random walks for 10, 100 and 1000 steps.

3.7) Instead of rnd.random() use the simple random generator $r_n = (a\ r_{n-1} + c)\ \%\ m$ which generates random number in the range 0 to m-1. How does the walk look like for r0=1, a=3, c=4, m=128? Also try m=129 and m=130 and some other values for all four parameters.

3.8) Generate many random walks with lengths in the range from 1 to 1000 and determine the mean squared end-to-end distance $\langle R^2 \rangle$ for each length. How does it depend on N?

3.9) A real polymer can not cross itself, i.e. different atoms can not occupy the same space. Generate self-avoiding random walks by storing all previously visited sites of the same walk and terminate and discard the walk when a previously visited is revisited. How does the fraction of succesful walks depend on N? What is the maximum value of N that you can reasonably consider? You can improve the algorithm somewhat by only generating moves in three directions, not back in the direction where you just came from. How does that improve the succes?

3.10) Compute the mean of the square of the end-to-end distances for the self-avoiding random walk for the range of N that you can cover. What is the difference with the normal random walk (suggestion: use log-log scale)?