2. Random processes & complex systems

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

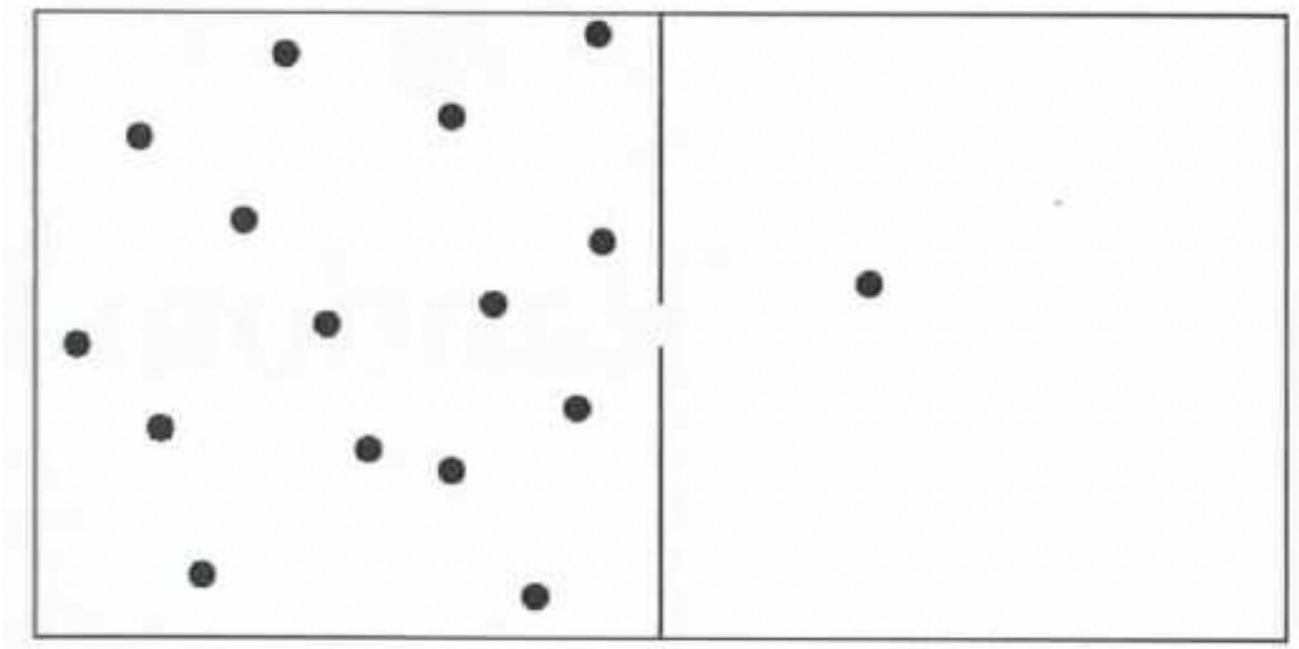
Random processes

- 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 inertia
- Langevin dynamics: dynamic, inertia

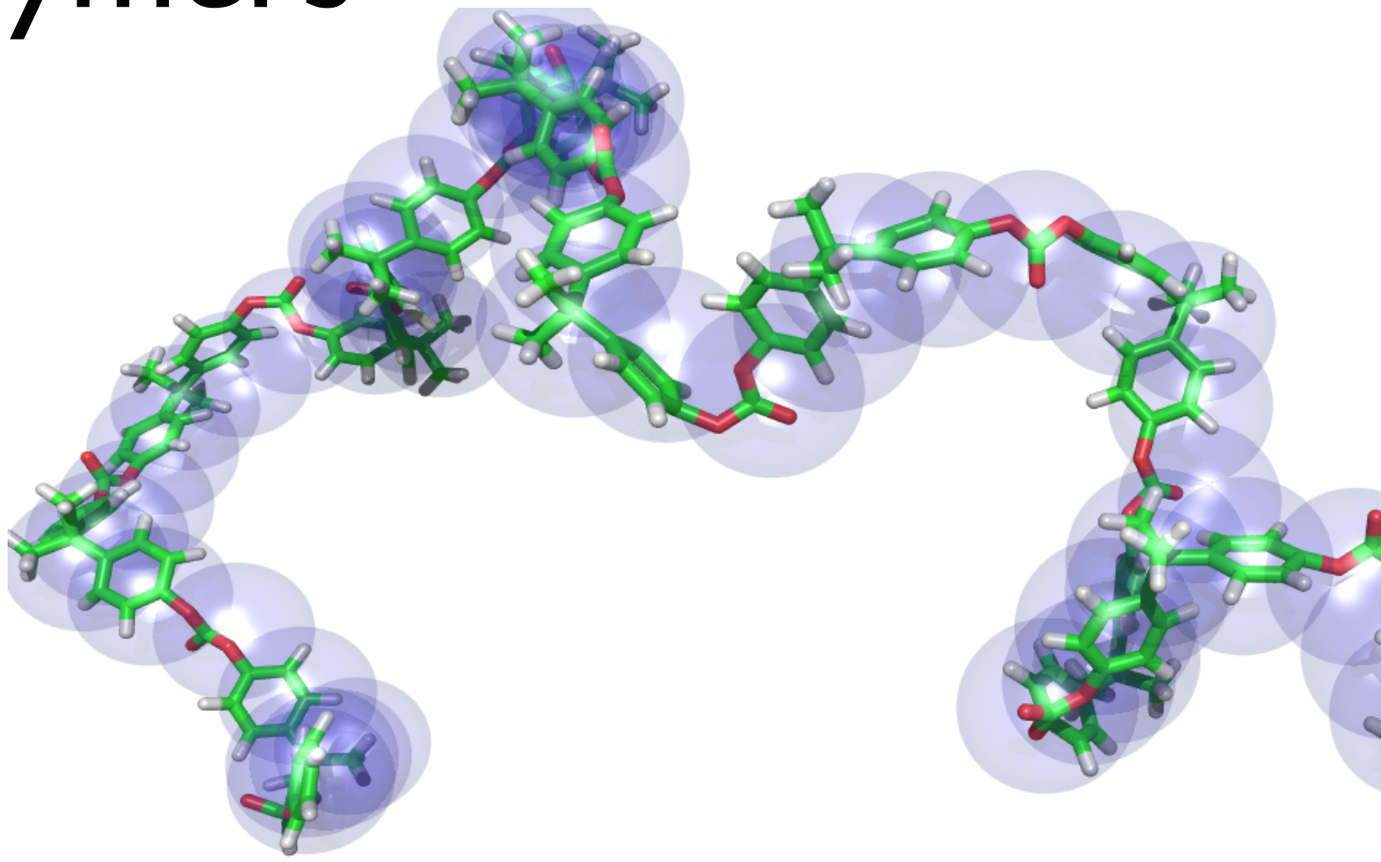
Particles in a box



- 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

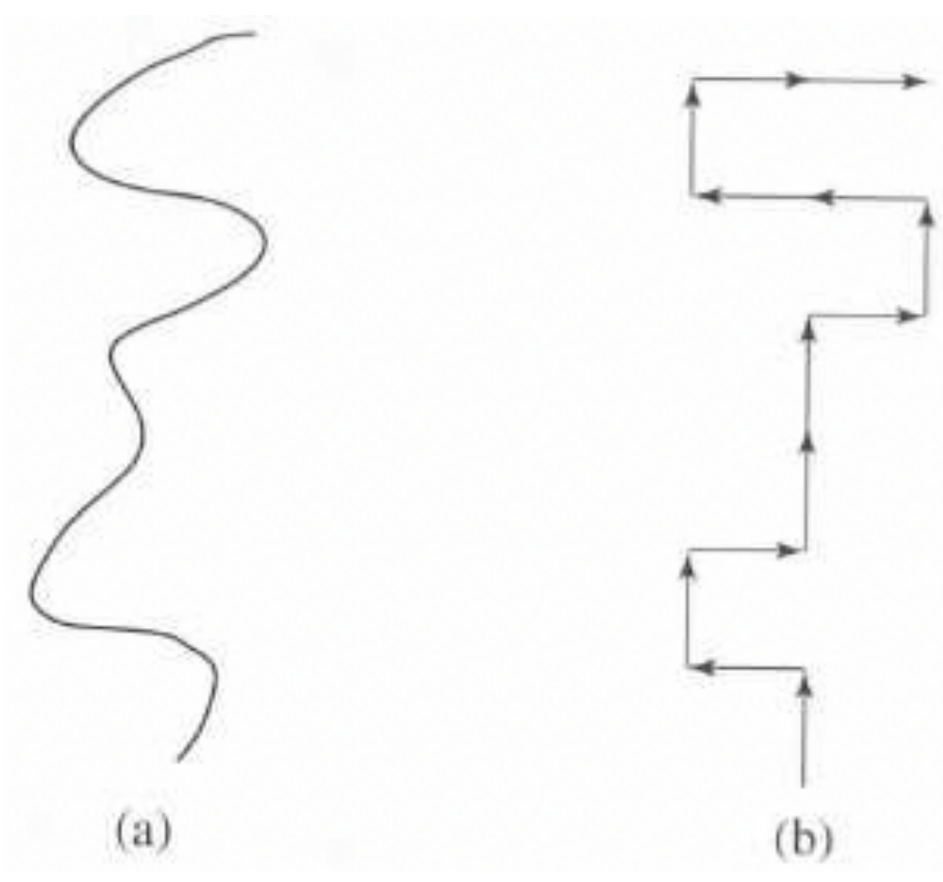
- Example: polycarbonate



- Polyelectrolytes, applications e.g.

“Ideal” polymer chains

- If parts of the polymer (and solvent) have very similar interactions, the polymers “feels” no difference in interaction in different conformations, this called:
  an **ideal** polymer
- You can approximate a polymer by a discrete random walk



Ideal polymer

- No potential energy differences:
  - **All conformations are equally probably**
    - For sampling conformations we need to sample all possible conformations with equal probability
      - This is what a random walk does
        - Each direction of a step is equally probable

Measuring polymers

- Polymers (can) adopt many different conformations
- An informative measure is the average "size" of a polymer
  - End-to-end distance: $\mathbf{R} = \mathbf{r}_N - \mathbf{r}_1$
    - Root mean square or second (central) moment R:

      first moment: $\langle \mathbf{R} \rangle = \mathbf{0}$, second moment: $\sqrt{\langle |\mathbf{R}|^2 \rangle}$

      - Compare with diffusion
- Experimentally one can measure the radius of gyration:

$$R_g^2 = \frac{1}{N} \sum_{k=1}^{N} |\mathbf{r_k} - \bar{\mathbf{r}}|^2$$

Self-avoiding random walk

- In dense polymer melts iparts of the chain do not occupy the same space ⇒use a self-avoiding random walk

- Important: when a random walk revisits a point, you have to reject the walk and generate a new one

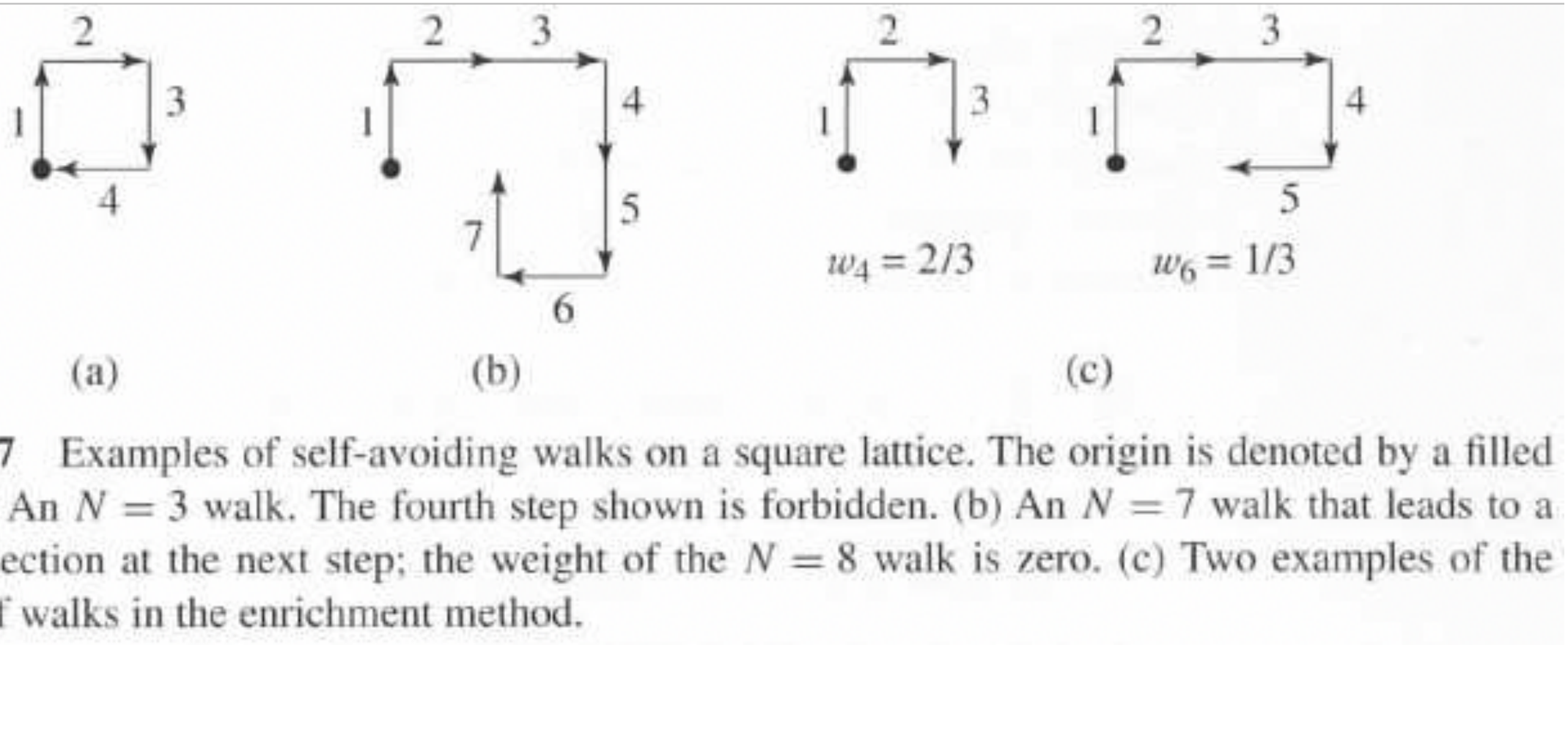


Figure 7.7 Examples of self-avoiding walks on a square lattice. The origin is denoted by a filled circle. (a) An $N = 3$ walk. The fourth step shown is forbidden. (b) An $N = 7$ walk that leads to a self-intersection at the next step; the weight of the $N = 8$ walk is zero. (c) Two examples of the weights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

Generating SAWs faster: 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

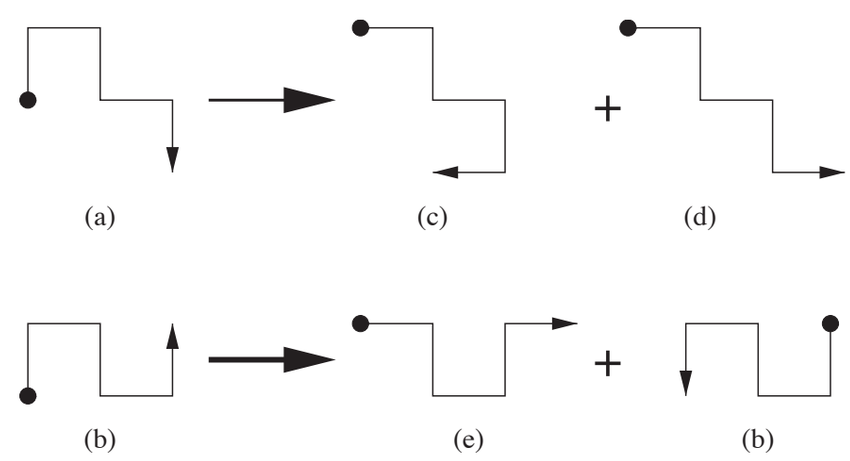


Figure 7.9: The possible transformations of chains a and b. One of the two possible transformations of chain b violates the self-intersection restriction and the head and tail are interchanged.

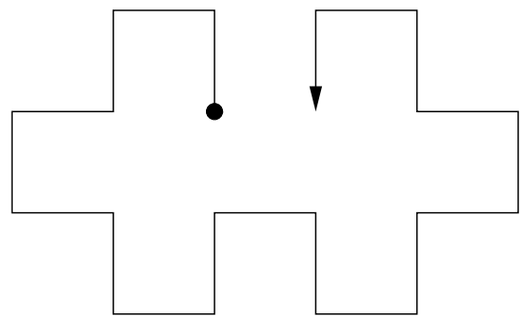


Figure 7.10: Example of a double cul-de-sac configuration for the self-avoiding walk that cannot be obtained by the reptation method.

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 $1 \leq x_i, y_i \leq L$. 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.

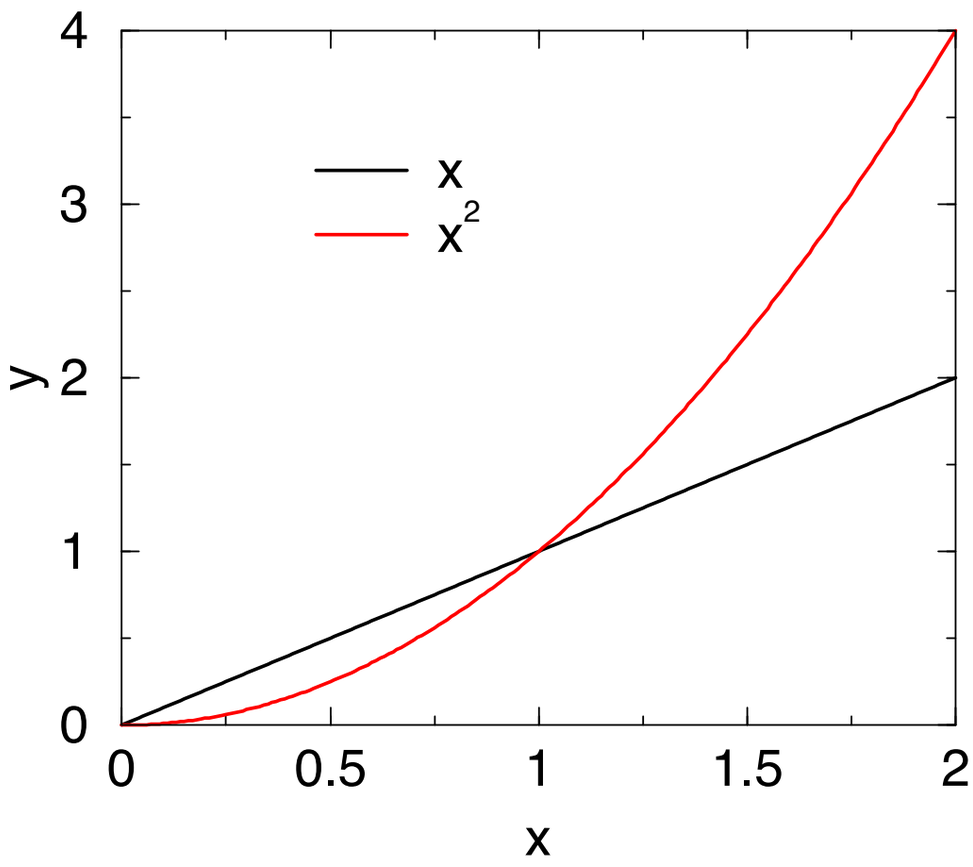
Getting accuracy from “random” simulations

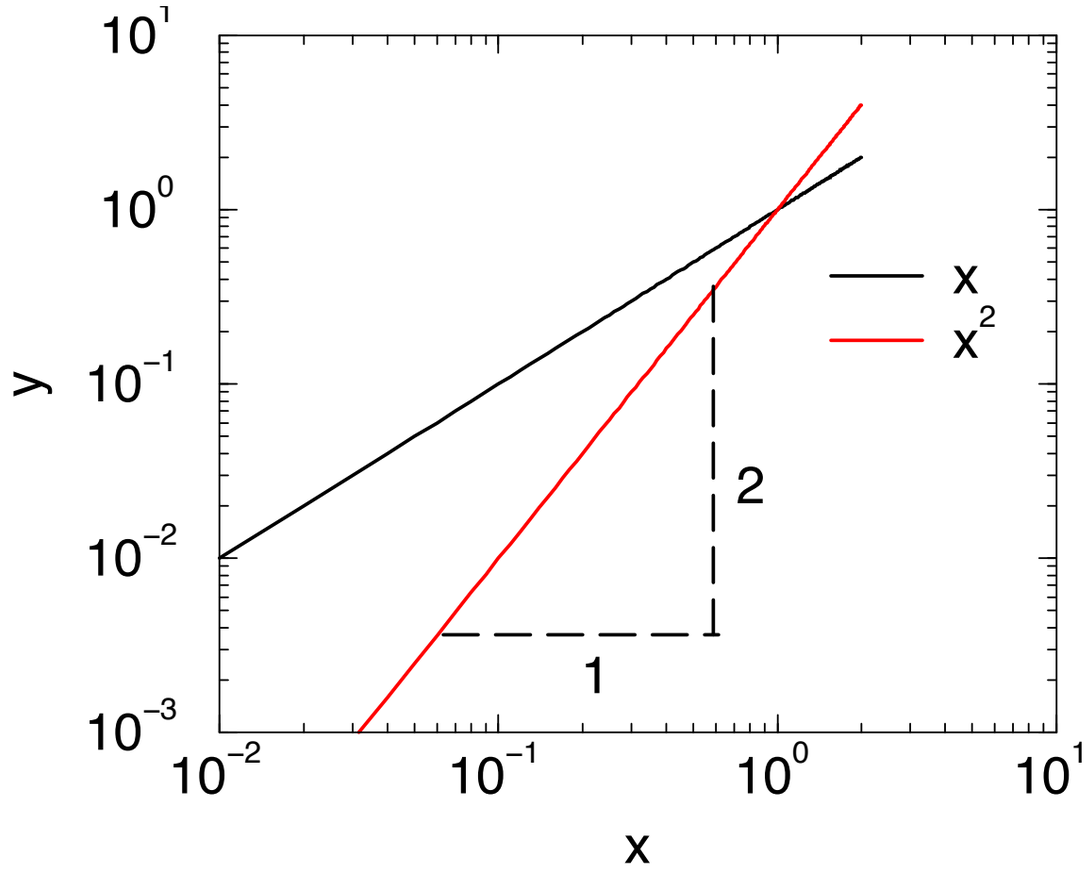
- …

Estimating fluctuations & errors

- Given M samples of a quantity R:
- The variance: $\mathrm{Var} = \langle R^2 \rangle - \langle R \rangle^2$
- Estimate of the root mean square fluctuation: $\sqrt{\mathrm{Var}\,\frac{M}{M-1}}$
- Standard error estimate: $\sqrt{\frac{\mathrm{Var}}{M-1}}$

Log-log scale





References

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

Complex systems

Physics

- Reducing natural phenomena to a few simple laws

- In this lecture:
  cellular automata introduced by Neumann and Ulam in 1948

- More complex systems (not in this lecture):
  - neural networks
  - genetic algorithms
  - growing networks
  - …

Cellular automata

1. Space is discrete and consists or a regular array of sites.
2. The rule for the new value of a site depends only on the old values.
3. Time is discrete. The variables at each site are updated simultaneously based on the values of the variables at the previous time step. Hence, the state of the entire lattice advances in discrete steps.

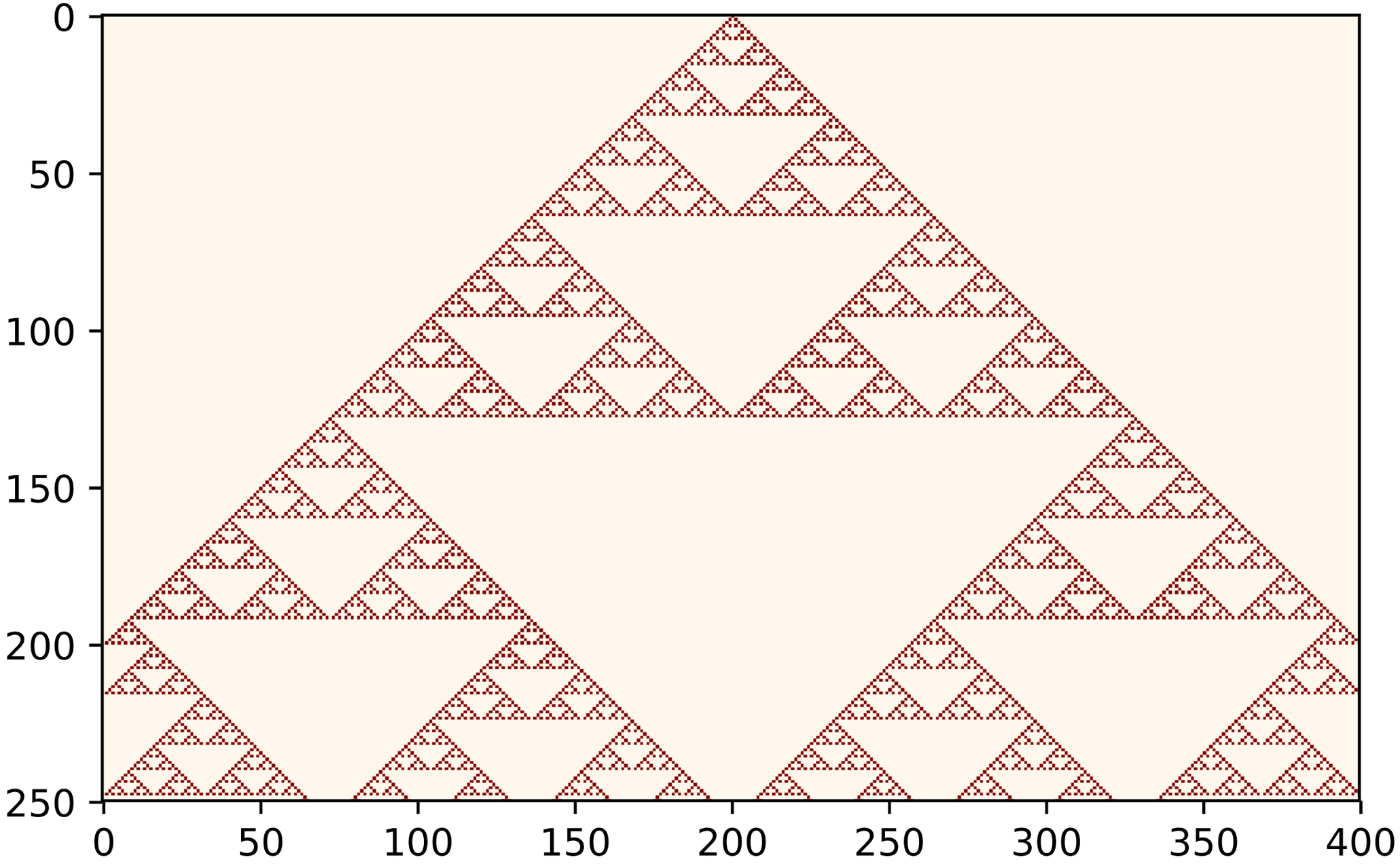
1D Cellular Automaton

t	111	110	101	100	011	010	001	000
t+1	0	1	0	1	1	0	1	0

- This is rule 90 = 01011010, $90 = 2^1 + 2^3 + 2^4 + 2^6$

- All $2^8 = 256$ 1D cellular automata have been cataloged
(Wolfram, 1984)

Rule 90



Elementary cellular automata

- Wolfram: http://mathworld.wolfram.com/ElementaryCellularAutomaton.html

Game of life

- Setup:
  - infinite two-dimensional grid of square cells
  - each cell: live or dead
  - every cell interacts with its eight neighbors
- Rules:
  1. Any live cell with fewer than two live neighbors dies, as if caused by under-population
  2. Any live cell with two or three live neighbors lives on to the next generation
  3. Any live cell with more than three live neighbors dies, as if by overcrowding
  4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction

Continuum traffic models

- Longitudinal Traffic model: Intelligent-Driver Model (Dresden)

$$\frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*}{s}\right)^2\right]$$

$$s^* = s_0 + \left(vT + \frac{v\Delta v}{2\sqrt{ab}}\right)$$

- parameters:
  - desired velocity when driving on a free road, v0
  - desired safety time headway when following other vehicles, T
  - acceleration in everyday traffic, a
  - "comfortable" braking deceleration in everyday traffic, b
  - minimum bumper-to-bumper distance to the front vehicle, s0
  - acceleration exponent, δ.

Cellular automaton traffic model

- Integer arrays for cars position and velocity: x_i, v_i,
  i indexes a car, not a lattice site

- Rules, in this order and loop over all cars for each rule separately:

  1. If $v_i < v_{max}$, increase the velocity vi of car i by one unit, that is, $v_i \rightarrow v_i + 1$.
     This change models the process of acceleration to the maximum velocity.

  2. Compute the distance to the next car d. If $v_i \geq d$, then reduce the velocity to $v_i = d - 1$ to prevent crashes.

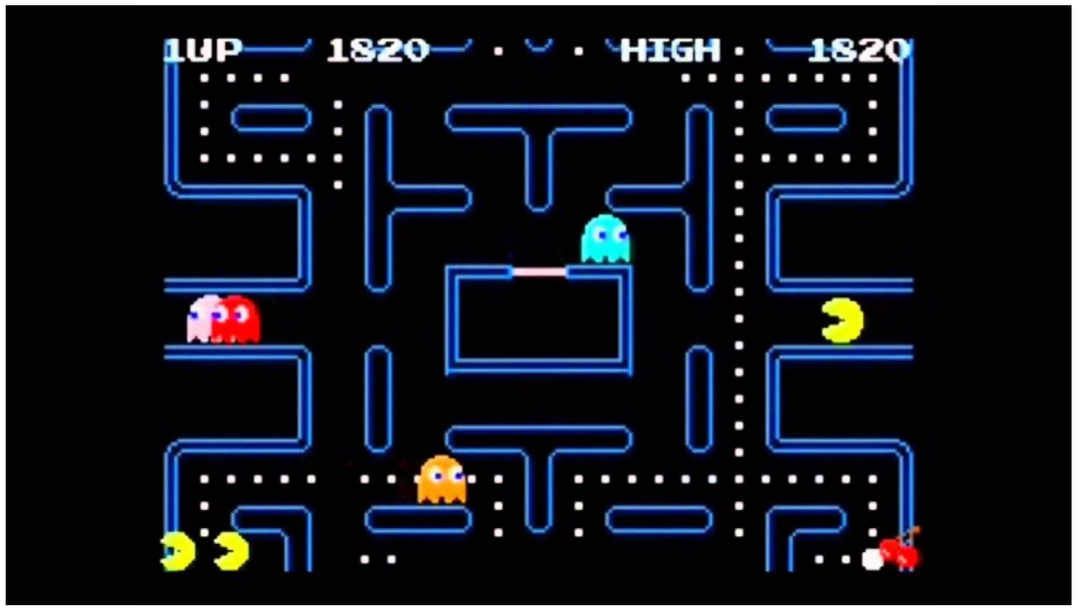
  3. With probability p, reduce the velocity of a moving car by one unit: $v_i \rightarrow v_i - 1$, only do this when $v > 0$ to avoid negative velocities

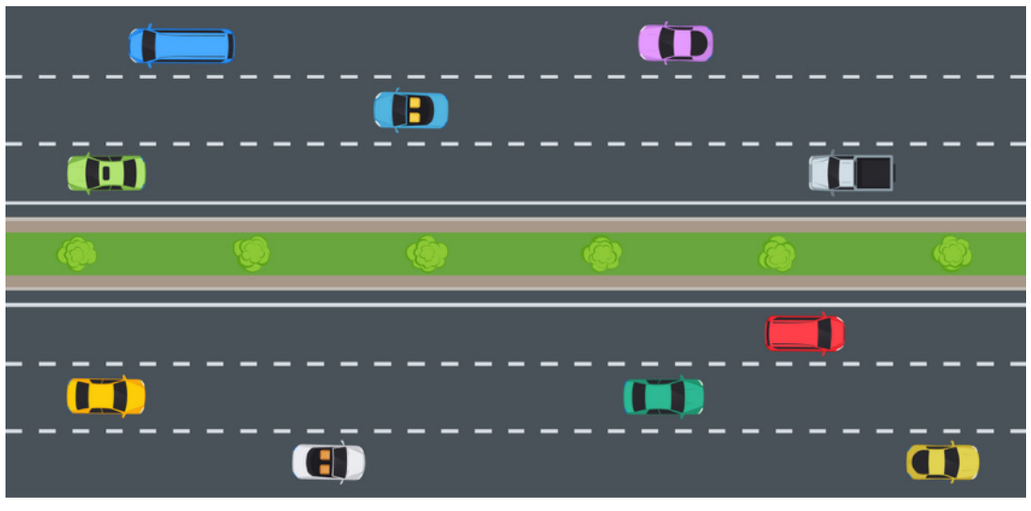
  4. Update the position x_i of car i so that $x_i(t + 1) = x_i(t) + v_i$

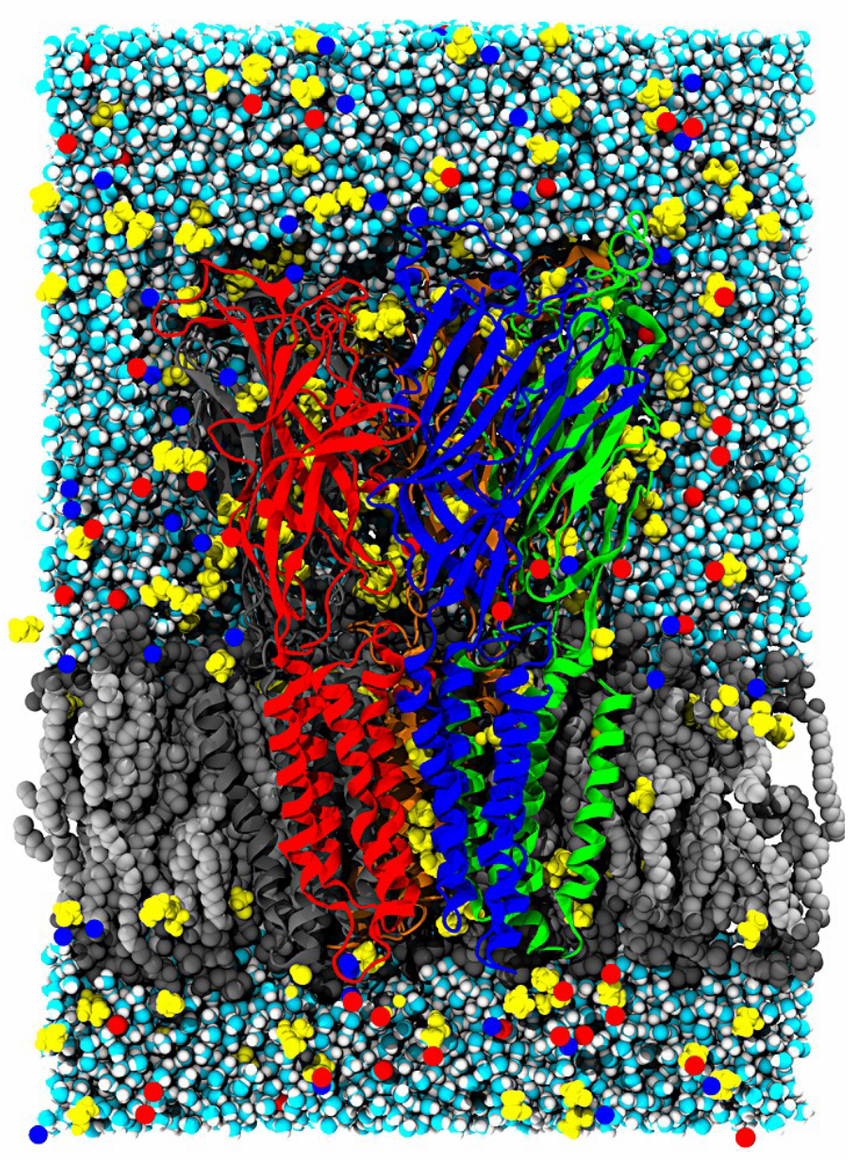
- Probability p add some randomization

- Flow rate: sum of the cars v_i's divided by the length of the road

Periodic boundary conditions







In python use the modulo operator % for computing distances:

d(i, i+1) = (x[(i + 1) % numCars] - x[i]) % roadLength

Traffic modeling at KTH

- https://www.kth.se/ctr/research/model-development/mezzo-mesoscopic-traffic-simulator-1.726113

Self-organized critical phenomena

- Large, rare events: earthquake, avalanche, stock market crash
- Cause:
  - special set of circumstances?
  - or part of a more general pattern?
- System is critical if:

$$(1) \quad N(s) \sim s^{-\alpha}$$

- where N(s) is the number of events of size s
- Contrast with combining random events:

$$(2) \quad N(s) \sim e^{-\left(\frac{s - s_0}{\sigma}\right)^2}$$ (characteristic scale σ)

Sand piles & avalanches

- Critical phenomena, example: https://www.youtube.com/watch?v=Jb7bsEFcpNs

Forest fire model

- Simple forest fire model on a L × L grid
- Update rules:

1. Randomly grow new trees at time t with a small probability g from sites that are empty at time t − 1
2. A tree catches fire due to lightning with probability f
3. Trees on fire ignite neighboring trees, which in turn ignite their neighboring trees, etc. The spreading of the fire occurs instantaneously.
4. Trees on fire at time t die (become empty sites)

Projects: Polymers as 2-dimensional random walks

2.1 a) 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.

2.1 b) Instead of `rnd.random()` use the simple random generator $r_n = (ar_{n-1} + c) \% m$ which generates random number in the range 0 to m-1. In Python you can use int(4*r/m) to convert r to an integer between 0 and 3. How does the walk look like for r_0=1, a=3, c=4, m=128? Also try m=129 and m=130 and some other values for all four parameters.

2.1 c) Switch back to `rnd.random()`. Generate many random walks for a few, not all 1000, values of the step count in the range from 1 to 1000 and determine the root mean square end-to-end distance $\sqrt{\langle R^2 \rangle}$ for each step count How does the end-to-end distance depend on N? Also plot the standard error estimate.

Projects: Polymers (continued)

2.1 d) 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?

2.1 e) Compute the mean squared 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)?

Projects: traffic model

2.2 a) Implement the cellular automaton traffic model with periodic boundary conditions. Note that you need to follow the exact order of steps given for the automaton. Run it for parameters $v_{\max} = 2$ and $p = 0.5$. (Plotting suggestions: plot the road on the x-axis, time on the y-axis and dots for the cars; for a movie plot symbols running along the edge of a circle (don't use radians for the simulation though)) Allow the system to equilibrate before recording the flow rate. Run with different car densities (number of cars / road length). Plot the flow rate versus the density. This plot is called the fundamental diagram. Explain its qualitative shape. At what density do traffic jams begin to occur?

2.2 b) Now we will estimate the statistical accuracy. Take a road length of 50, use 25 cars, $v_{\max} = 2$ and $p = 0.5$. Run multiple simulations, for each collect the flow rate averaged over 100 time steps. Compute the standard error esimate of the flow rate. How many simulations do you need to get a standard error of 0.001? How long does the equilibration need to be to get accurate results? Does the equilibration time depend on how you choose the initial conditions?

Projects: traffic model (continued)

2.2 c) When do the results, i.e. the fundamental diagram, become independent of the size of the system (at constant density)? Or conversely, when shortening the road length with constant density, when do the results start to deviate from those of long road lengths? Be aware that longer roads need longer equilibration times.

2.2 d) Now we will also look at fuel consumption. We assume that the fuel consumption of a car is $1 + v + 0.2\,v^3$, where the constant 1 is for the inefficiency/stationary running of a petrol engine or the heating of an electric car, the linear term for the rolling resistance and the cubic term for the air resistance. Run simulations for a car density of 0.4 with $p = 0.25$ for $v_{\max} = 1, \ldots, 5$. Plot, as a function of $v_{\max}$, the flow rate and the average fuel consumption summed over all cars divided by the flow rate. Can you explain the results?