Exercises: Simulation Methods in Statistical Physics

Astrid S. de Wijn

March 19, 2013

1 Exercises lectures 1–5 (due 18 Feb)

1.1 Lyapunov exponents of the Lorentz gas

Consider the two-dimensional random Lorentz gas. This is a set of randomly placed, hard, circular scatterers with a single point particle bouncing around between them. When the point particle bounces elastically off the scatterer, its velocity is reflected.

a) How many degrees of freedom does this system have? How many dimensions does the phase space have?

b) In this exercise, the aim is to calculate all the Lyapunov exponents of the system. These are the possible rates at which infinitessimal perturbations in phase space can blow up or shrink after long time. There are as many Lyapunov exponents as the phase space has dimensions. Show that for any Hamiltonian system, including this one, the sum of all Lyapunov exponents is 0.

c) Show that there are two Lyapunov exponents in the two-dimensional Lorentz gas that are 0. The easiest way to do this is to come up with two examples of perturbations that will not grow or shrink exponentially. Use the result from b) to obtain a relation between the largest and smallest Lyapunov exponents.

The largest positive Lyapunov exponent can be calculated by considering the growth of an arbitrary infinitessimal pertubation that is not one of the two perturbations you have found in c). After long time, the growth of this perturbation will be dominated by the largest Lyapunov exponent. Consider two nearby paths with the same absolute velocity, but slightly different direction. The difference between the two trajectories can be described by the relative difference in angle $\delta\theta$. The largest Lyapunov exponent can then be written as

$$\lambda^+ = \lim_{t\to\infty} \frac{1}{t} \log(\delta\theta(t)/\delta\theta(0)) \ . \tag{1}$$

One could obtain the Lyapunov exponents numerically by simulating a trajectory and calculating this expresssion. At low densities, however, it is not necessary to resort to numerics.

d) During a free flight between collisions, the relative angle of the velocities does not change. Consider the growth of the difference in position between the two trajectories during a free flight, and calculates how this affects the collision normal for the next collision. Show that the difference in angles after a collision, $\delta\theta'$, can be written as

$$\delta\theta' = (1 + 2l/(a\cos\phi))\,\delta\theta \ , \tag{2}$$

where ϕ is the angle between the velocity and the collision normal, a is the radius of a scatterer, and l is the length of the free flight. Low density, i.e. $\rho a^2 \ll 1$, means that for the mean free path l, $l/a \gg 1$. Thus, in the term between parentheses, the 1 can be neglected.

e) Explain that this Lyapunov exponent can be calculated as

$$\lambda^+ = \nu\langle \log(2l/(a\cos\phi)\rangle \ , \tag{3}$$

with ν the average collision frequency and $\langle\rangle$ the averaging over the distribution of l and ϕ.

f) In a Monte-Carlo simulation of this system, you would also need to know these distributions. You would have to draw collision parameters for the next collision with a scatterer from these distributions. Show that the probability densities are given by

$$P(l) = \frac{1}{l}_{mf} \exp(-l/l_{mf}) \ , \tag{4}$$

$$P(\phi) = \frac{1}{2}\cos(\phi), \text{with} -\frac{\pi}{2} < \phi < \frac{\pi}{2} \ . \tag{5}$$

Here, l_{mf} is the mean free path between two successive collisions. Show also that $l_{mf} = 1/(2\rho a)$ and $\nu = v/l_{mf} = 2\rho v a$.

g) To get the final result, you will have to integrate over these distributions. If you were doing a Monte-Carlo simulation using the above distributions, instead of an analytical calculation, you would have essentially used a Monte-Carlo method to calculate that integral. Monte-Carlo methods are often used to numerically calculate integrals. Do this particular integral manually and prove that

$$\lambda^+ = 2\rho a v \left(\ln(1/\rho a^2) + 1 - \ln 2 - C\right) \ . \tag{6}$$

Here C is Euler's constant, $C = 0.577215...$

1.2 Integrating equations of motion

The details of the Runge-Kutta algorithms and midpoint rule can be found in Thijssen's appendix, page 570.

a) Show that the RK2 algorithm is not symplectic, by calculating the jacobian for a general Hamiltonian system and demonstrating that, in general, phase-space volume is not conserved.

b) RK2 and RK4 are shown in the book and explained using midpoints. That is to say, the second order version uses two mid points, k_1 and k_2, and the fourth order one uses k_1, k_2, k_3, and k_4. Choose some k_1, k_2, k_3 that you find convenient. Derive a third-order RK3 algorithm, by Taylor expanding and finding the constants c_1, c_2, and c_3 such that $x(h) = x(0) + c_1 k_1 + c_2 k_2 + c_3 k_3 + O(h^4)$.

c) When you use a time-step size that is much too large, for example when you simulate the Argon fluid in the programming exercise, there will be a drift in the energy. Explain why, on average, the energy will increase.

d) Why is it not recommended to integrate Hamiltonian equations of motion with a Runge-Kutta method? If you end up doing this anyway, for whatever reason (probably lazyness), what should you do to make sure that your simulations produce valid results?

1.3 Hangman: an exercise in entropy and resource scaling

Entropy is a central concept in statistical physics. It is not only useful when dealing with large numbers of particles, but also in other areas dealing with large numbers of something or other. This hangman exercise is an example of that.

Hangman (swedish hänga gubben) is a game where you must guess a word of specific length by guessing letters in it. If you guess too many wrong letters, you are hanged and "die". If you guess a correct letter, the game will reply by telling you where in the word it is. An example transcript of a game:

```
________.
guess e
______e_.
guess n
______e_ (n).
guess s
______e_ (ns).
guess r
______er (ns).
guess i
```

```
______er (nsi).
guess o
_o____er (nsi).
guess a
_o____er (nsia).
guess m
_om___er (nsia).
guess computer
The word was: computer.
```

Suppose you are playing a computerised version of hangman. The hangman selects a random word of length w from a dictionary that contains N words in total. You want to create your own hangman player, which has access to the same dictionary, and has to come up with the best letter to guess at every step. This may not be a physics problem exactly, but, because the dictionary is large, it is a computational and statistical problem, and the same things that we consider in this course play a role.

a) Every word (or state) is equally likely to be selected by the hangman computer. Define an entropy S in the same sense as in statistical mechanics, based on the number of possible words n at any particular stage in the game.

b) Entropy is all about information. What does the entropy you have defined in a) tell you about the information that you have about the word?

c) Start simple with $w = 3$ where you have not made any guesses yet. Write down an expression for the expectation value of the change in the entropy $\langle \Delta S \rangle$ when your first guess is "a". You can use the notation of $W(s)$ to denote the number of words that fit a particular value s of the observable. In this case s can take the string values a__,_a_,... aaa,___(a). The latter denotes the state of the observable when there has turned out to be no "a" in the word. Denote the set of all states of the observable for the next step after ___ that contain an "a" by A(___,a).

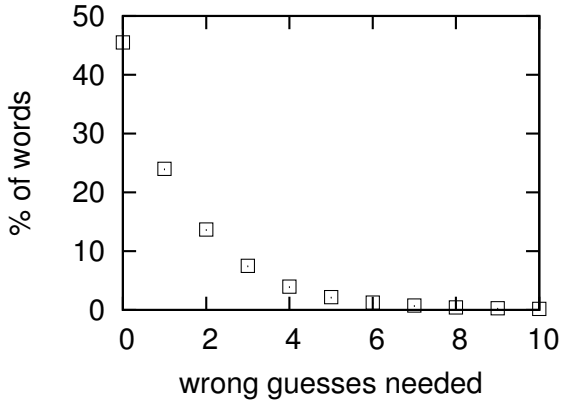
d) At every step, you obviously want to guess a letter that will give you a lot of information, thus minimising the entropy, while making as few mistakes as possible. Write down an expression for the average entropy change per mistake, $\langle \Delta S \rangle / \langle m \rangle$. Use $A(s,\mathrm{b})$ to describe the set of all possible replies starting from the state of the observable s with, for example, a "b" guessed correctly.

e) Before you could implement this idea, you would also need to know if this is computationally doable. What order of magnitude are the parameters that will play a role in the cpu and memory resource scaling: word length w, dictionary size N, and alphabet length a?

f) How would you implement this? Describe an algorithm explicitely. Try to minimise the number of times you have to search through the dictionary. Make a few simple assumptions about languages that seem reasonable and estimate roughly how the evaluation of the parameter that is to be optimised scales with N, w, and a. As this depends on how you choose to implement the expression found in e), it may not be the same for everyone.

g) Do you think this computation will be doable?

For your information, the algorithm I have in mind was able to correctly guess 46% of words from a dutch dictionary without any mistakes, and 97% with 5 or fewer mistakes. See the plot below.


50
40
30
20
10
0
% of words
0
2
4
6
8
10
wrong guesses needed

2 Exercises lectures 6–8 (due 25 Feb)

2.1 Langevin dynamics for thermostatting

Find the Fokker-Planck equation somewhere in the literature (even Wikipedia will do). For a particle subjected to langevin dynamics, the probability density is governed by the FP equation.

a) Write down the langevin equation for a point particle in a potential well in one dimension.

b) From the langevin equation, argue why the 2×2 diffusion tensor in the FP equation for this particle has only one nonzero element. Argue that this element does not depend on the position or velocity.

c) Show that for a one-dimensional particle in a potential well, langevin dynamics produce a probability density equal to the canonical probability distribution. Langevin dynamics thus produce a suitable thermostat.

2.2 Frenkel and Smit exercises

Answer questions 8 and 9 on pages 58 and 59.

2.3 Drawing a variable from a non-uniform distribution

Random number generators usually produce homogeneously distributed random numbers. Often this is not really the distribution you want. Suppose you have at your disposal a random number generator which produces random numbers x distributed uniformly between 0 and 1.

a) Using a method similar to the one described in the lecture for Gaussian random numbers, devise a way to obtain a random number y distributed between 0 and ∞ according to the density

$$\rho(y) = \frac{1}{l} \exp\left(-\frac{y}{l}\right) \ , \tag{7}$$

where l is some positive constant. Prove that your scheme gives the correct distribution. For the Monte-Carlo simulation of the random Lorentz gas, which we have used as an example in the lecture, you would need this distribution for the free-flight times (see exercise 1.1).

b) The other distribution in the Lorentz-gas exercise was

$$\rho(\phi) = \frac{1}{2} \cos\phi, \text{with} -\frac{\pi}{2} < \phi < \frac{\pi}{2} \ . \tag{8}$$

There are other ways to get nonuniform distributions, for instance by first drawing from a uniform distribution, and then rejecting the outcome with some probability that depends on the value of the random number. Find such a way to draw from the distribution in Eq. (8).

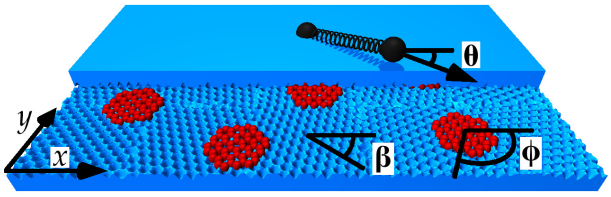
3 Exercises lectures 9–10 (due March 6th)

3.1 Applying thermostats sensibly

When you perform MD simulations in the canonical ensemble, or where for some other reason temperature must be controlled with a thermostat, it is important to consider what sort of dynamics would be realistic.

a) Suppose you would like to know if you can turn your Argon simulation into a crystalline solid. To achieve this, you would have to cool it down very slowly, changing the temperature of the system slowly. This means you would have to apply a thermostat. Would you use the Nosé-Hoover thermostat or a stochastic thermostat like the Langevin thermostat? Why?

b) Consider another simulation of a non-equilibrium system. The setup would be similar to the figure below (taken from my own work). Two parallel rigid surfaces (blue) are made of regularly placed atoms. There are other particles (red) embedded between the two surfaces, keeping them seperated. These could be single atoms, or larger molecules (like in the figure). The interactions between the red particles and atoms in the blue surfaces is governed by some sort of effective interaction, for example Lennard-Jones. The surfaces are kept parallel and orthogonal to the z-axis. One is kept completely fixed. The other surface is forced to slide with respect to the first. This will inject energy into the system. Without any additional measures, the system would heat up. How would a real system cool down? In the simulation a thermostat is necessary to model this. How would you thermostat this system and why?



c) Consider the same system as in b), but now with several layers of particles between the two surfaces. Suppose the particles cannot move from one layer to the other. How would you thermostat this system and why?

3.2 Importance sampling in the Ising model

a) Consider, for example, the three-dimensional Ising model. How many possible states does the system have if it has $N \times N \times N$ sites? Make some simple guesses about computer speeds and the number of operations needed to calculate the terms in the sum. Give an estimate of the order of magnitude of time it would take to numerically calculate the partition function for $N = 5$ with today's computers.

$N = 5$ is nowhere near the thermodynamic limit. Obviously, for large systems, it is not a good idea to consider every state. We can get around this problem by using me thods which estimate thermodynamic properties by sampling a small subset of all configurations. Suppose we attempt to estimate the partition function by randomly sampling B states, and calculating their contributio n. Let α_1 through α_k be the results of the sampling. Consider the estimator for a quantity X

$$X_{\text{estimate}} = \frac{\frac{1}{B}\sum_{k=1}^{B} X(\alpha_k)\exp[-\beta E(\alpha_k)]}{\frac{1}{B}\sum_{k=1}^{B}\exp[-\beta E(\alpha_k)]} \,, \tag{9}$$

where $E(\alpha)$ and $X(\alpha)$ are the energy and relevant quantity respectively.

b) Argue that, if $B \gg 1$, the expectation value of X_{estimate} is equal to the thermal average of X.

c) Recall what happens to the Ising model at low temperatures. What happens to the terms in the sums in equation (9)? How many states (compared to the total number) would you have to sample to get the results to converge to something sens ible when the temperature approaches 0?

The reason for the difficulties with the estimator in equation (9) is that all states are equally likely to be sampled, even though some hardly contribute to the partition function. The variance of the terms in the sum is large. Suppose instead that we sample from the range of possible states with some probability $\rho(\alpha)$ that depends on the state α.

d) Write down an estimator for the quantity X, making sure that the expectation value of the estimate is equal to the thermodynamic average. The estimator should be of the form

$$X_{\text{estimate}} = \frac{1}{B}\sum_{k=1}^{B} \xi(\alpha_k) \ . \tag{10}$$

e) The right choice of $\rho(\alpha)$ can greatly improve the convergence of the sum. So far in this problem, $\rho(\alpha)$ has been constant. By calculating the variance of $\xi(\alpha)$, show that $\rho(\alpha) \propto \exp[-\beta E(\alpha_k)]$ is an efficient choice of probability density that works well for physical quantities.

3.3 Metropolis

While we know we wish to sample the distribution according to the Boltzmann factor, we do not know the normalisation constant. In Markov-chain Monte-Carlo methods, rather than just selecting a random configuration, a new configuration is generated from an old one in a random way, thus producing a string of pseudo-random states.

Let the associated transition probability from configuration α to configuration α' be denoted by $P(\alpha \to \alpha')$. If the transition probabilities are chosen appropriately, then the states are distributed according to the correct ensemble distribution. Two conditions need to be satisfied for this to be the case, accessibility and detailed balance.

a) Write down the condition for detailed balance and substitute the equilibrium distribution.

b) Use induction to show that if the accessibility and above detailed balance conditions are satisfied, the distribution of elements of a long Markov chain approaches the correct distribution.

The Metropolis algorithm is a commonly used implementation that achieves this. The first step is to randomly generate a new state from the old one (trial move). In the Ising model for instance this could be by flipping a random spin. The state is then accepted with some probability. The transition probability from the old state to the new one in the Metropolis algorithm is:

$$P(\alpha \to \alpha') = \begin{cases} & \text{if} \quad E(\alpha') < E(\alpha) \\ \exp[-\beta(E(\alpha) - E(\alpha')] & \text{if} \quad E(\alpha') > E(\alpha) \end{cases} \ . \tag{11}$$

In a practical implementation one would generate a random number between 0 and 1, and accept the new state if the random number is less than the transition probability.

c) Show that this algorithm satisfies the detailed balance condition you have obtained in a), and that in the above described scheme for the Ising model, it satisfies the accessibility assumption as well.

4 Exercises lectures 11-14 (due March 20th)

4.1 Trial moves and ensembles in MC

Solve questions 12 and 13 on pages 135 and 136 of Frenkel and Smit.

4.2 Variational QMC

a) Derive equation (12.13) on page 378 of Thijssen.

b) Do exercise 12.5 a, b, and e on page 420 of Thijssen.

4.3 Path integral QMC

Do exercise 12.2 on pages 418 and 419 of Thijssen.