



Übung 8: Temperature and motions

Contents

1. Background	1
2. Energy model.....	2
3. Distribution.....	3
4. To Do.....	4
4.1.1. For everybody	4
4.1.2. For people who like programming.....	5
4.1.2.1. Hints.....	5
4.1.3. For people who do not like programming.....	6

1. Background

It has been claimed or will be claimed that different kinds of energy surfaces have different sensitivity to temperature. As you heat a system, it might move with a higher frequency. Alternatively, the frequency might stay the same, but the amplitude will increase. The point of this Übung is to simulate this.

In the lectures, the arguments were based on macroscopic objects like a pendulum. Here we will think in terms of atoms on a scale of Å. On this scale we often use a different model for motion. If you think of a pendulum, you think of inertia. You push an object and it keeps

moving. On the atomic scale, this is less helpful. You push an atom and it almost immediately has a collision.

We use a very simple model for movements. Our particle attempts a random move - as if it had been hit by another particle. We calculate the energy and, if the energy has become too high, we forget the move. It never happened. If the energy goes up a bit, we might accept the move as described in section 3.

2. Energy model

We are going to look at an energy surface with two minima, separated by a barrier. We do not have a physical model for $U(x)$, the energy as a function of a coordinate x . Instead, we invent a formula with the shape we want. This will be two minima separated by a barrier. The atom cannot move far to the left or right, since it would hit other atoms. A form for this would be

$$U(x) = x^4 + cx^2 \tag{1}$$

For aesthetic reasons, we can add a zero offset and say

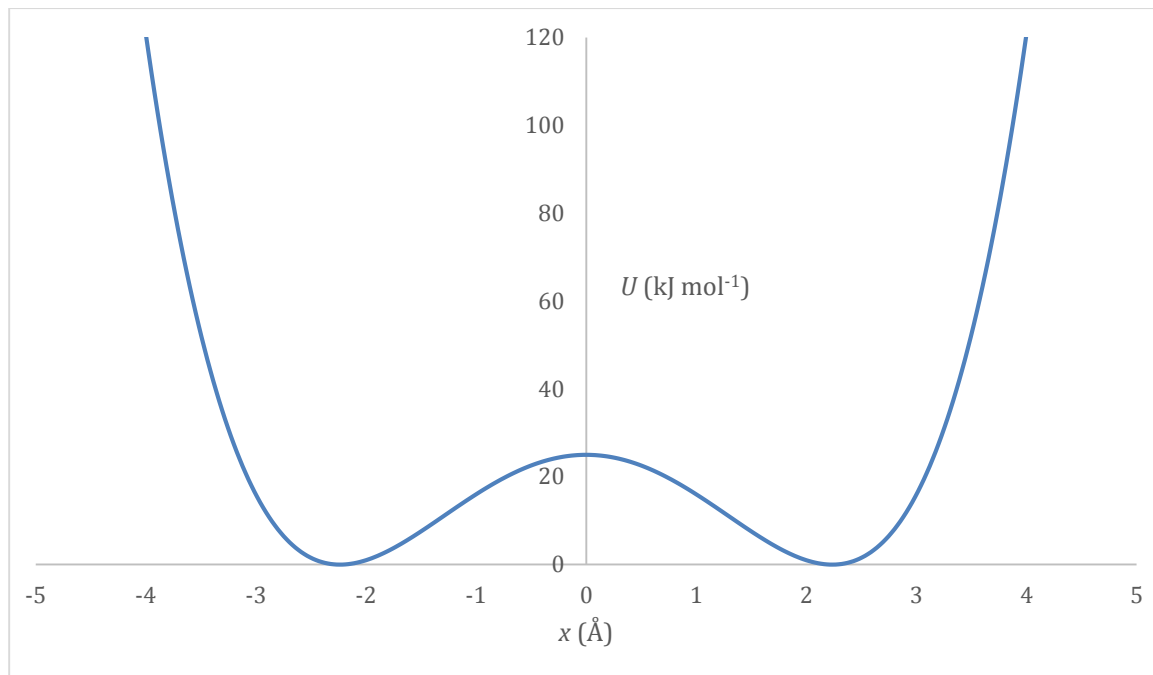
$$U(x) = x^4 + cx^2 + d \tag{2}$$

Only the constant c is really important. We will use $c = -10 \text{ kJ mol}^{-1} \text{ \AA}^{-2}$ and

$d = \frac{c^2}{4} = 25 \text{ kJ mol}^{-1}$ giving,

$$U(x) = x^4 - 10x^2 + 25 \tag{3}$$

This form is defined for all x and it is always positive which is pretty.



3. Distribution

You will not be able to avoid the Boltzmann distribution in the summer semester, so you can start to make friends now. Basically, we want an expression for the probability p_i of a state i . This depends on the energy $U(x_i)$

$$p_i = Z^{-1} \exp\left(\frac{-U(x_i)}{kT}\right) \quad (4)$$

There is a name for Z , but we will save it for next semester. T is the temperature and k is the Boltzmann constant. We choose our units so $k = 1$. You could almost use eq. 4 to calculate a probability of a state/coordinate i , except we cannot calculate Z^{-1} , so we make it disappear. If your system starts with the coordinate x_i and you want to know the relative probability of a new coordinate, x_j , you can calculate the relative probability from

$$\frac{p_i}{p_j} = \frac{Z^{-1} \exp\left(\frac{-U(x_i)}{kT}\right)}{Z^{-1} \exp\left(\frac{-U(x_j)}{kT}\right)} = \exp\left(\frac{-U(x_i) + U(x_j)}{kT}\right) \quad (5)$$

Let us define $k = 1$ and $\Delta U_{ij} = U(x_j) - U(x_i)$ then

$$\frac{p_i}{p_j} = \exp\left(\frac{-\Delta U_{ij}}{T}\right) \quad (6)$$

Now, we have the ingredients necessary to sample from the correct distribution. We start with some coordinate x_i . We make a trial move to x_{trial} and calculate the new energy. If the energy goes down, we accept the move. If the energy goes up, we calculate the probability of the new x from eq. 6 and accept the move with this probability. If the energy is much worse, ΔU will be a big positive number and $\exp(-\Delta U)$ will be close to zero. If the energy is a bit worse, ΔU will be small and positive, $\exp(-\Delta U)$ will be near to zero.

Note the effect of temperature. If the temperature is near to zero, it makes the probabilities smaller. It is harder for the system to go uphill in terms of energy. If T is large and positive (hot), it is easier for the system to accept a new state, even when the energy becomes a bit worse.

4. To Do

You can decide what to do.

4.1.1. For everybody

Our real aim is to look at the effect of temperature on the rate of barrier crossing. Look at the figure on page 3. Every time the system moves from $x \leq 0$ to $x > 0$ or vice versa, we count a transition. At the end of a calculation, we say our frequency is $\frac{n_{\text{transition}}}{n_{\text{acc}}}$ where n_{acc} is the number of steps which were accepted. In a monte carlo procedure, n_{acc} may be much much smaller than the number of attempted steps. To summarise the overall procedure

```

For some temperature  $T$ 
start with an arbitrary  $x_i$ 
for  $n_{\text{step}}$  steps
    try to move
    if (the move is accepted)
         $n_{\text{acc}}++$ 
        if  $x$  has changed left→right OR right→left
             $n_{\text{transition}}++$ 
print  $\frac{n_{\text{transition}}}{n_{\text{acc}}}$ 

```

and you have to repeat this for a set of temperatures.

Let us define frequency as $f = \frac{n_{\text{transitions}}}{n_{\text{acc}}}$, then the final result will be a plot of f as a function of T or f versus $\log T$.

When you are debugging or making a plot, you might want to try a few hundred steps.

When collecting data, you want a minimum of 10^5 steps.

4.1.2. For people who like programming

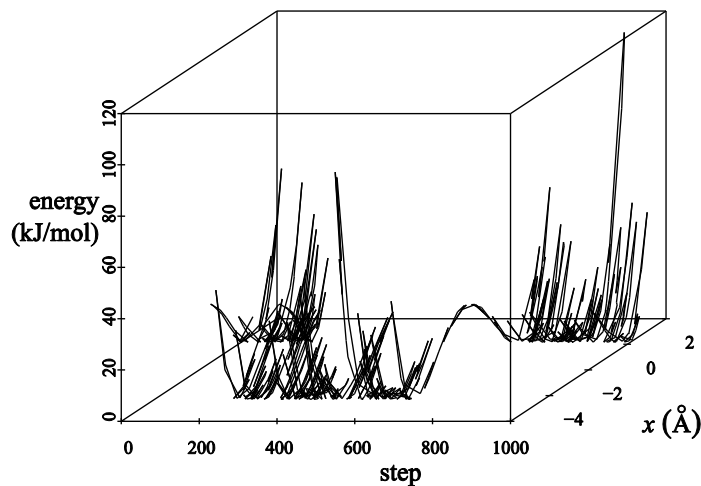
Use any language you want. Speed is not an issue.

- write a program that implements the idea of the box on page 4.
- Plot frequency, f , versus $\log T$ for about 10 values from 2, 20, 50, 100, 200... 1000
- Send your source code and the plot to hansen@zbh.uni-hamburg.de by 19. Jan 2015.

4.1.2.1. Hints

It is easy to make mistakes.

- Random number function
Do not write your own random number generator. Use a function built in to the language you use. We do not need a high quality random number generator. In C, one normally uses `random()` or `drand48()`.
- Really random numbers
Do not seed the random number generator with the time of day or something that looks a bit random. In all simulation problems, you want to be able to reproduce the results. Best practice is to have a default seed with an option to set the seed.
- Parameters
For our simulation, the values of c and d in eq. 2 do not really matter, but if you use these values, we know what to expect.
- Step size
Given the values of c , we know that a reasonable size for the random steps is in the range -0.5 to 0.5.
- Testing
If you were to plot out the path of the system, it should look like this:



The system should spend most of its time in the valleys and occasionally jump from one side to the other. To check the program, you should have an option to print out the x -coordinate and energy at every step or after every accepted step.

Check if the number of accepted steps increases when you raise the temperature.

4.1.3. For people who do not like programming

Have a look in `~torda/uebung_freq_mc`. There is a program which will run a calculation and print out minimal answers. There is a brief man page which one can read with a command like, `man ~torda/uebung_freq_mc/man/mc_freq.1`.

- Copy the `Makefile` and C code from `~torda/uebung_freq_mc` to a directory in your account. Compile the code (`make mc_freq`).
- After reading the man page, you should see how to set up a run with different temperatures. Do runs with temperatures in the range of 10 to 500. For each run, use 10^6 points.
- At low temperatures, the number of accepted moves can be quite small. The statistics are not very reliable. Repeat the runs for a temperature of 20, but with different random number seeds (`-s` option) or with really random numbers (`-u` option). If you do 10 runs, what is the variation in the value of n_{acc} ?
- At a temperature like $T = 20$, transitions are not very common. Use the `-r fname` option to plot out a trajectory of the particle. Use your favourite plotting program, but with a program like `gnuplot`, you would just have to say,
`plot 'fname' using 1:2.`