**Monte Carlo simulations for the estimation of molecule pair interaction**

**TASK 1**

Once you obtained the average number of dimers, in order to compare with the experiment you have to compute the percentage of dimers with respect to the total number of molecules (50 for coverage 0.02 that could also be estimated as 2\*(average number of dimers) + average number of molecules in the hypothesis that there are no clusters.)

The simulation with the specified parameters could provide an average number of dimers of e.g. 2.45

To get a rough estimate of the requested quantity it is sufficient to remember that kB\*T at room temperature is ~1/40 eV

So

E=0.025 eV \* 200K / 300K \* ln((2.45/50)/0.3) = -0.03 eV

**TASK 2**

You will not get a result close to the experimental finding. This indicates that the physics underneath is not governed by nearest neighbor interactions. Indeed to explain the experimental result long range coulombic repulsion as well as short range van der Waals interactions had to be included. (This explanation was of course beyond the aim of the exercise)

The aim of **Task 3** was just to understand that, at a given temperature, increasing the binding energy (more negative values) results in increasing the size of clusters (with a consequent reduction of their number). Also increasing the coverage will result in larger clusters.

Given a value of the coverage and of E, if we increase the temperature we increase the probability for a molecule to separate from a cluster thus reducing the average size of clusters and increasing their number.

**Task 4** is very important: if we remove the condition for the detailed balance (fulfilled through the recipe of Metropolis) and substitute it with “if enew < eold” we will of course do something that has nothing to do with ensemble averages, we will miss a link with temperature and the simulation will simply stuck in the attempt to reach a geometry corresponding to a global minimum for the energy.

**Task 5**

“Can you imagine ...” No or maybe yes, I would use something related to neighbors lists. The goal of this task was to encourage you to look in the literature (or in the web) for solutions to this problem. More simply it is important that you know that many “topological problems” in atomistic simulations find brilliant solutions in the literature, it is instructive to think a possible solution but one should always compare with the efficiency of known solutions. In this case I was playing with the “image” library of “scipy” that solves the problem but you can also find in wiki The Hosen-Kopelman algorithm.

<https://en.wikipedia.org/wiki/Hoshen%E2%80%93Kopelman_algorithm>

**Task 6** yes it would be correct. Of course one has to create moves in a correct way avoiding overlap between molecules (but this is also true when moving one molecule at each step). What will happen is that every move will involve a “bigger” change in energy thus the acceptance rate will become lower. On the other hand the moves will be more “efficient” so, summing up, the best efficiency in the simulation can be tuned selecting an optimal amount of molecules to move at each step and, also one could decide a maximum “size” for the displacement of a molecule.

**Kinetic Monte Carlo simulations for the diffusion of molecules on a substrate**

**Task 1 and 2**

Simulations that differ for the overall height of the barriers will show a different behavior in the time evolution that is governed by a Poisson law. The time evolution depends exponentially on the barrier height and linearly on the attempt frequency. This aspect can be understood also quite intuitively (the example I provide is quite exaggerated so try to find more “correct” examles): suppose in one simulation we have events such as “freezing the lake of Zurich”, and in another simulation we have events such as “moving a mountain”.

At each KMC step one event has to happen (otherwise it means our planet is gone). Now suppose you live inside the simulation and you don’t have a clock, you studied quite a lot of history and in one case you observe the town of Zurich realizing that from time to time it happens that the lake is freezing, you would guess that, on average, 50-200 years are separating each event. In the other case you observe mountains, simply appearing and disappearing, you will guess that time is running at intervals of some hundred millions years..

In addition to this, the main difference between task 1 and task 2 is that in tassk1 the diffusion barrier and the binding barrier are the same while in task 2 the binding barrier is higher compared to the diffusion barrier.

In the first case as soon as two molecules “get in contact” they have a high probability to bind (almost the same as diffusing), they will not be able anymore to move. The simulation will stop as soon as no more events will be possible (this will usually happen well before 100k steps) and the molecule will have formed “dendritic” like structures.

If the barrier for the binding event will be higher compared to the barrier for diffusion, the molecules will have the possibility to explore more “convenient” binding geometries before irreversibly binding. This will result in a much longer simulation (more steps are needed before all molecules bind) and the final arrangement of the molecules will be characterized by clusters with more regular shape.

**Task3-4**

The most delicate point in setting up a KMC simulation is the ability to identify a limited set of “simple” events (like diffusion of atoms, binding events, exchange between surface and subsurface atoms, breaking of bonds…) through which the most relevant physical/chemical aspects of the system we want to describe can be reproduced.

“simple” means that it should be easy to transform the event into elementary instructions inside the code and that the update of the list of events will be not challenging after an event has been executed.

Of course I can include in my simulation every event for which I am able to define:

i)an attempt frequency to escape from an initial configuration towards a final configuration

and

ii)the corresponding energy barrier I have to cross (transition state)

but the more “long range” will be the effect of my event the more difficult will be to code it efficiently and to update efficiently the list of events at each simulation step.

**Task5**

No, in general, for the KMC algorithm described in the lectures it is not possible to apply two events at the same time. We can have complex events (still corresponding to a transition state and an attempt frequency) but I cannot apply two events at the same time. The problem would be mainly defining how time should evolve, and also handling of two simultaneous events that were not meant to be simultaneous: it could be that two events are possible at one step but they are not possible at the same time because, for example, they would bring two different particles into a same empty lattice point (the event A->C exist in the list of events, also the event B->C exists,

but we cannot do both at the same time)