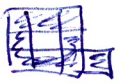
Modelling crystal growth

- growth is relatively compliated process, because several different stochastic processes happen simultaneously (capture of a particle on the surface, diffusion of particles on the surface, evaporation of a particle)
- it can proceed differently on different surfaces, it depends if particles are captured from gas or liquid etc.
- it is often difficult to find out how particular processes contribute to the growth, therefore we use simplified stochastic models based on Monte-Carlo simulations and study their behaviour to determine which processes are relevant
  - it turned out that some models behave similarly and they fall to several classes that can be assigned also to more complicated systems
- there are discrete models on lattices and continuous models
  - here we demonstrate the growth process on two rather simple models in one dimension and introduce also the kinetic Monte-Carlo method using this process, but it has much broader use

Discrete models of growth

- particles can be only in the nodes of a lattice
- basic three processes:
  - deposition of new particles
  - evaporation of particles from the surface
  - migration (diffusion) of particles on the surface

- not all these processes are used in every model
- they happen with probability that depends on the current configuration of the surface

- the simplest models are called <u>solid-on-solid</u> (SOS) models that do not consider „overhangs"  or „cavities" 
  - particles lie on top of each other and the surface is characterized by the height of columns of particles above the base layer

- to describe the surface quantitatively during simulations we can determine various mean quantities as functions of time as for example (in 1D case)

<u>mean height</u> $\bar{h}(t) = \frac{1}{L} \sum_{i=1}^{L} h(i,t)$ where L is the lattice size

<u>roughness of the surface</u> (variance of h)

$$w(t) = \left(\frac{1}{L} \sum_{i=1}^{L} \left[h(i,t) - \bar{h}(t) \right]^2 \right)^{1/2}$$

or <u>height correlation function</u>

$$G(r,t) = \frac{1}{L} \sum_{i=1}^{L} \left[h(i+r,t) - h(i,t) \right]^2$$

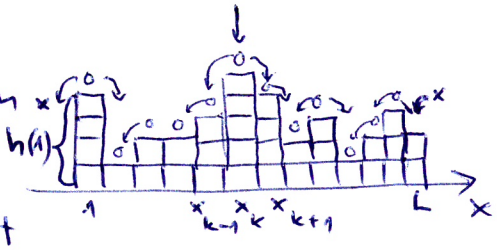
- in the simple simulations when we only add particles

$$\bar{h}(t) \propto t$$

- for other quantities, dependence on time can be $\propto t^{\beta}$ and for finite systems, there can be a change in the exponent after „saturation" of the surface
- quantities can depend also on the lattice size as L^{α}

Examples: the basic simplest models of growth

1) Edward-Wilkinson (EW) model



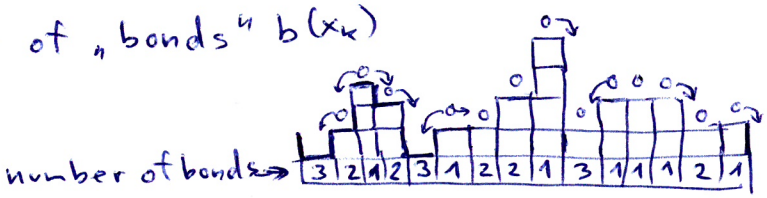
- a simple model of sedimentation
- if a particle (randomly) falls to the site x_k then if the height $h(x_k) \leq h(x_{k-1})$ and $h(x_{k+1})$ then it stays there

otherwise it moves (just once) to the neighbouring site with the smallest height or if $h(x_{k-1}) = h(x_{k+1}) < h(x_k)$ then randomly to the left or to the right

- periodic boundary conditions

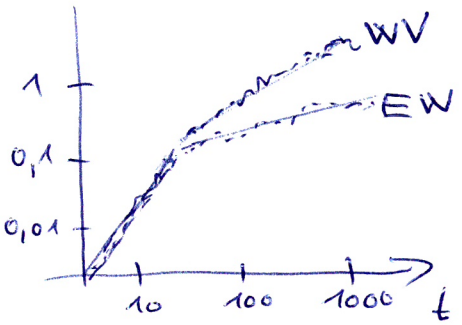
2) Wolf-Villain (WV) model

- similar as EW-odel but the choice of the site where a particle finally deposits is given by the number of „bonds" $b(x_k)$



- the particle goes where $b(x_k)$, $b(x_{k-1})$ or $b(x_{k+1})$ is largest

- they give different exponents for $w(t)$ or $G(r,t)$ after saturation



Kinetic Monte Carlo - illustration on growth models

- various processes happen with different rates R_α
- for example in growth models we can have
  1) different rates for deposition R_1, R_2, R_3
     depending on the number of neighbouring particles
     (R_i means the number of particles deposited per second to the site with i neighbours)
  2) rates of evaporation R_4, R_5, R_6 - also depend on the number of neighbours
  3) rates for migration (diffusion) - $R_7, \ldots, R_{N_p}$
     - depending on neighbourhood / total number of possible processes
- all rates R_α determine the model and in principle, these could be calculated from the first principles, but that is usually difficult => estimations or from experiments
  - they usually depend on the environment (like temperature etc)
- individual processes are realised by elementary events (such as deposition to a specific site etc.) and their number n_α depends on the current configuration A_k in the k-th iteration of the simulation (for example if we fill in a hole in A_{k-1}, the neighbour can't migrate there in A_k)
- because R_α are not normalized, to determine probabilities of particular elementary events we have to divide R_α by the total rate $R(A_k) = \sum_{\alpha=1}^{N_p} n_\alpha(A_k) R_\alpha$

- then, a certain elementary event of the process with rate R_α happens with probability

$$\Pi_\alpha = \frac{R_\alpha}{R(A_k)}$$

- a possible, but very inefficient algorithm of kinetic MC simulation could be

1) choose an event that can happen for A_k
2) generate $\xi \in \langle 0, 1 \rangle$
3) if $\xi < \Pi_\alpha$ then implement the event and update all necessary variables

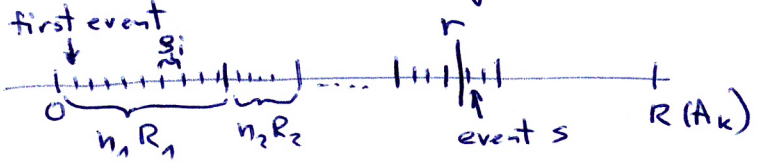
   otherwise do nothing
4) repeat

- the problem is that for many processes and possible events Π_α will be too small (or for small R_α also) and we get too many rejections

- n-fold way or BKL algorithm (rejection-free algorithm)

(according to Bortz, Kalos and Lebowitz: J. Comp. Phys. 17 (1975) 10)

- an algorithm that always chooses an event and with correct probability
- now we generate a random number $r \in \langle 0, R(A_k) \rangle$ and we determine an event corresponding to r



- algorithm
  1) generate $r \in \langle 0, R(A_k) \rangle$
  2) find event s using $\sum_{i=1}^{s} g_i \geq r$ ← the smallest such s, $g_i = \begin{cases} R_1 \text{ for events of process 1} \\ R_2 \\ \vdots \end{cases}$
  3) implement the event
  4) update $R(A_k)$, $n_\alpha(A_k)$ etc.

in general $\mathcal{O}(N)$ → where N is the number of all events

also $\mathcal{O}(N)$ → but often faster

- to make it faster, we can first choose a process

using $\sum_{\alpha=1}^{\sigma} n_\alpha(A_k) R_\alpha \geq r$ where σ is the smallest such index

and then we choose an event for the process

(this can be trivial as often all events for a specific process have the same rate)

- but we still have to update all $n_\alpha(A_k)$

- more general algorithm for a lot of processes and events

could even group some processes first and choose a group, then a process and finally an event

to optimize the number of operations to find an event

Time in kinetic MC simulations

- in standard MC simulations, the number of steps (iterations) does not relate to physical time and generated configurations do not correspond to real evolution of the system

- in kinetic MC, the situation is different because changes of configurations correspond to real processes and events that can happen with corresponding rates

- therefore, it is convenient to introduce time into such simulations using the total rate $R(A_k)$, particularly in each step we increase the time by a random interval given by

$$\Delta t_k = -\frac{1}{R(A_k)} \ln \xi$$

where $\xi \in \langle 0, 1)$ is uniformly distributed random number

and thus, the total time after n steps is $t = \sum_{k=1}^{n} \Delta t_k$

and mean value of quantity X is $\langle X \rangle_{\langle 0,t \rangle} = \frac{1}{t} \sum_{k=1}^{n} \Delta t_k X(A_k)$

- to justify the expression for Δt_k, we can start from an approximative solution of the master equation for Markovian processes for which the change of probability $P(A)$ of finding the system in the configuration A in time t is govern by

$$\frac{\partial P(A,t)}{\partial t} = -\underbrace{\sum_{A' \neq A} W(A \to A') P(A,t)}_{\text{the change from the state } A} + \underbrace{\sum_{A' \neq A} W(A' \to A) P(A',t)}_{\text{the change into the state } A}$$

where $W(A \to A')$ gives probability of transition from A to A' per unit of time, in general dependent on time (it is not the same matrix as in equilibrium simulations)

- in kinetic MC simulations, we choose $W(A \to A')$ in such a way to correspond to the correct behaviour of the system in time by considering N_p processes with prescribed rates R_α resulting in $W(A \to A') = F(R_1, \dots, R_{N_p})$

- if, in the master equation, the second term is negligible (we consider only changes from the configuration A) we have

$$\frac{\partial P(A,t)}{\partial t} = -R(A,t)\, P(A,t)$$

where

$$R(A,t) = \sum_{A' \neq A} W(A \to A', t) \doteq \sum_{\alpha=1}^{N_p} n_\alpha(A_k) R_\alpha = R(A_k)$$

↑ approximation made in the kinetic MC method

is the total rate of change

- for short times, we can write approximately

$$P(A, t+\Delta t) = \underbrace{e^{-R(A_k)\Delta t}}_{\text{this term expresses the probability that nothing happens during the interval } \Delta t} P(A, t)$$

(the larger is R, the faster decrease towards zero)

- under assumptions that there is one elementary event at a time with the total rate R and that it is a Poisson process (events are independent), the probability that something happens during time t is $1 - e^{-Rt}$. To this cumulative distribution function, the following probability density corresponds

$$g(t) = R e^{-Rt}$$

$$\left(\text{as } \Phi(t) = \int_0^t R e^{-Rt} dt = R\left[-\frac{1}{R} e^{-Rt} \right]_0^t = 1 - e^{-Rt} \right)$$

- therefore, we generate a random time interval after which some event happens in such a way to obtain the corresponding prob. density $g(t)$, using the inverse of the cumulative distribution function $\Phi(t)$ as

$$\Delta t = \Phi^{-1}(v) = -\frac{1}{R} \ln(1-v)$$

where v is a random number from $\langle 0, 1)$ with the uniform distr.

- because $1-v$ is also a random variable on $\langle 0, 1)$ with a uniform distribution, we can use a simpler formula

$$\Delta t_k = -\frac{1}{R(A_k)} \ln \xi_{\langle 0,1 \rangle}$$

for the increment of time during the k-th iteration of kinetic MC simulation