



Friedrich-Alexander-Universität  
Faculty of Engineering

Department Materials Science

WW8: Materials Simulation

## **Practical: Kinetic Monte Carlo Simulation**

Leon Pyka (22030137)

**Supervision: Dr. Frank Wendler**

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	General Algorithm of the rejection free KMC Method . . . . .	2
<b>2</b>	<b>Diffusion Problems Solved with KMC</b>	<b>3</b>
2.1	Diffusion in a Cu thin film with time dependent diffusion rate . . . . .	3

## 1 Introduction

The Monte Carlo method broadly refers to a large toolset of modelling methods utilizing the generation random numbers to solve problems. The underlying algorithm is the Metropolis algorithm. As opposed to Molecular Dynamics Monte Carlo methods do not follow the simulated particles trajectories over, but randomly perturbed the system to asses the the resulting systems properties in comparison to the preceding system. The *Kinetic Monte Carlo* (KMC) method incorporates kinetic features to the simulation. Each transition from state to state is seen is a diffusive jump and can happen at time-scales larger than in Molecular Dynamics (which is limited by the time-scale of atomic vibrations) [1].

### 1.1 General Algorithm of the rejection free KMC Method

We start with a system of possible states  $i$  and possible transitions  $i \rightarrow j$ . The transition rate is  $\nu_{ij}$ . The following algorithm is reproduced from the problem statement [2].

1. start with state  $i$  at time  $t$
2. evaluate total jump rate  $\Gamma$

$$\Gamma = \sum_j \nu_{ij} \quad (1)$$

3. evaluate probability of jump by comparing  $\nu_{ij}$  to the total jump rate (eq. 2a) and calculate the cumulative probability 2b

$$p_j = \frac{\nu_{ij}}{\Gamma} \quad (2a)$$

$$P_j = \sum_{j' < j} p_{ij'} \quad (2b)$$

4. generate a random number  $R_1 \in [0, 1)$
5. chose state which satisfies  $P_{ij} = \min P_{il}, P_{il} > R_1$  (where  $P_{il}$  is a list ordered in descending order, to make sure that high probability steps are more likely to be executed)
6. generate random number  $R_2 \in [0, 1)$  and calculate jump time  $t_{ij}$  as

$$t_{ij} = -\frac{\ln(R_2)}{\Gamma} \quad (3)$$

7. update system configuration:  $i \rightarrow j$  and update time  $t = t + t_{ij}$
8. restart loop

## 2 Diffusion Problems Solved with KMC

### 2.1 Diffusion in a Cu thin film with time dependent diffusion rate

In this task we consider a sandwich structure of a Cu film and a Cu-10% Al film, both of thickness 100 nm. As Cu is an fcc-metal it is possible to simplify the set-up to a 2D simple cubic lattice. The diffusion coefficient is given as:

$$D = 1.49 \cdot 10^{-7} \exp\left(-\frac{136.1\text{kJ}}{RT}\right) \frac{\text{m}^2}{\text{s}} \quad (4)$$

In 3D-the jump rate for e.g. Carbon at octahedral interstitial sites in  $\alpha$ -iron is given as [3]:

$$\nu_{ij} = \frac{6D}{b^2} \quad (5)$$

so for a 2D-simple cubic with 4 possible moving directions it should be given as:

$$\nu_{ij} = \frac{4D}{b^2} \quad (6)$$

The jump rate  $\nu_{ij}$  can be derived from D as follows:

$$\nu_{ij} = \frac{4D}{b^2} \quad (7)$$

with  $b = 2.54\text{\AA}$ . As we are considering only one kind of diffusion process  $\nu_{ij}$  is the same for all jumps so eq. 1 simplifies to:

$$\Gamma = n\nu_{ij} \quad (8)$$

and the probability from eq. 2a simplifies as stated below:

$$p_{ij} = \frac{\nu_{ij}}{\Gamma} = \frac{1}{n} \quad (9)$$

where  $n$  is the number of jumps. In the simulation the temperature needs to be increased linear to the time - from 293 K to 600 K. So the temperature at every time step (after incrementation according to the algorithm in section 1.1) is found as:

$$T_i = T_{\text{start}} + \frac{T_{\text{fin}} - T_{\text{start}}}{t_i/t_{\text{total}}} \quad (10)$$

where  $T_i$  is the current temperature at time-step  $t_i$ ,  $T_{\text{start}}$  is the initial temperature,  $T_{\text{fin}}$  the temperature at the end and  $t_{\text{total}}$  the total time until the heating is finished. Actually running the simulation exposes some additional challenges. As the size of the time-step is inversely proportional to  $\Gamma$ . Meaning that the time-step at room temperature can become very large. So large indeed that it can exceed the total runtime of the simulation in the first step. To prevent this from happening the initially rejection free algorithm is updated to include the possibility of rejection. This is triggered when a certain time-step is exceeded. As the time and temperature are updated in the range of the time-step, but without performing a perturbation of the system. The second problem is the time-complexity of the simulation. As we have to make sure that the system is updated after every move, to prevent e.g. Al atoms from switching sites with other Al-atoms. If this is checked for every jump the time complexity instantly becomes of size  $O(n^2)$ . This can be tackled, by using a data-frame storing both the atoms coordinates and their neighbors'. In the initial implementation a `pandas.DataFrame` was used. The random moving direction was chosen from the neighbors-list using the `pandas.sample()` method. This resulted in a peculiar error - displayed in fig. 2.1. As all the atoms seem to have a preferred moving direction this might expose a problem with the pseudo-randomness of the `pandas.sample()` function. The tendency to choose the first moving direction (down) over the other direction shows that the the function seems not to be uniformly distributed, for the required sample size.

Fig. 2.1 shows a succesful implementation utilizing a `numpy.array` storing the individual atoms as objects carrying the type of the atom, the index of the neighboring atoms, the index of the sites with which the atom can switch places and the the number of switches the individual atom can perform. The flattening of the concentration gradient is qualitatively visible in fig. 3.

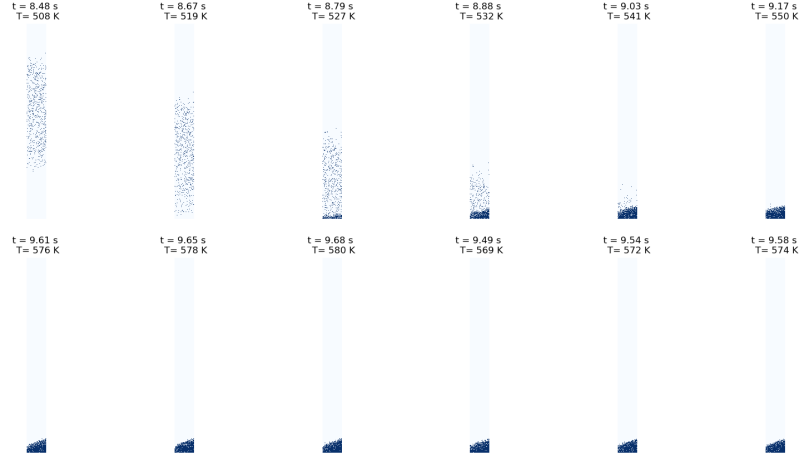


Figure 1: Failed implementation of diffusion in an  $\text{Cu}/\text{Cu-10\% Al}$ -sandwich structure. The blue dots represent Al-atoms at substitutional sites.

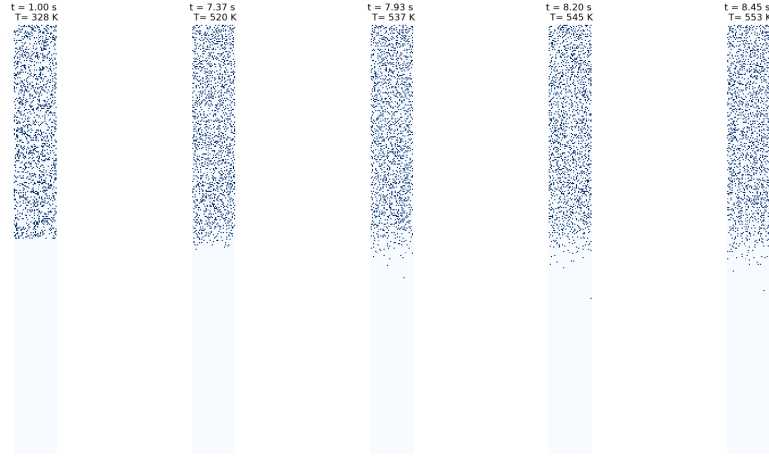


Figure 2: KMC-Simulation of a  $\text{Cu}/\text{Cu-10\% Al}$ -sandwich structure. The blue dots represent Al-atoms at substitutional sites.

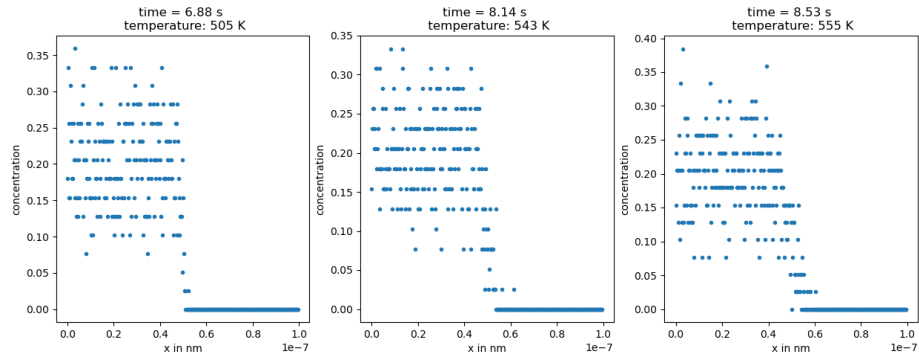


Figure 3: x-position against concentration of al per row at evaluated different times and temperatures.

## References

1. Voter, A. F. *INTRODUCTION TO THE KINETIC MONTE CARLO METHOD* in *Radiation Effects in Solids* (eds Sickafus, K. E., Kotomin, E. A. & Uberuaga, B. P.) (Springer Netherlands, Dordrecht, 2007), 1–23. ISBN: 978-1-4020-5295-8.
2. Zaiser, M. *Kinetic Monte Carlo Simulation - Problem Statement*
3. Gottstein, G. in *Physical Foundations of Materials Science* (ed Gottstein, G.) 155–196 (Springer, Berlin, Heidelberg, 2004). ISBN: 978-3-662-09291-0. [https://doi.org/10.1007/978-3-662-09291-0\\_6](https://doi.org/10.1007/978-3-662-09291-0_6) (2024).