Master’s thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

Marthe Strand Haltbakk

Kinetic Monte Carlo simulation of the early precipitation stages in Al-Mg-Si alloys using Cluster Expansion methods for energy barrier modelling

Master’s thesis in Applied Physics and Mathematics
Supervisor: Jesper Friis
Co-supervisor: Inga Gudem Ringdalen
June 2021

NTNU
Norwegian University of
Science and Technology

Marthe Strand Haltbakk

Kinetic Monte Carlo simulation of the early precipitation stages in Al-Mg-Si alloys using Cluster Expansion methods for energy barrier modelling

Master's thesis in Applied Physics and Mathematics
Supervisor: Jesper Friis
Co-supervisor: Inga Gudem Ringdalen
June 2021

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

NTNU
Norwegian University of
Science and Technology

Abstract

In this master thesis Al-Mg-Si alloys are studied using kinetic Monte Carlo simulations governed by energy barriers in order to gain insights in the precipitation process. The work is a continuation of earlier works that modelled the barriers from cluster expansion on activation energies from density functional theory (DFT) calculations. This work is expanded by implementing a kinetically resolved activation barrier model that ensures the preservation of the total energy, and maintaining detailed balance during simulation by explicitly including the total energy difference in the expression. The two methods are both trained on the same training data, and simulated using a system of 25 × 25 × 25 unit cells in a face centred cubic lattice structure containing 0.77% Si and 0.67% Mg at room temperature. The new method introduced an RMSE of 38.71 meV compared to 9.14 meV for the original method using the same training data. None of the methods are seen to form clusters during simulations, but the large drift in energy introduced in the earlier works was corrected by the new method. A third model barrier seen to form clusters from earlier works has been tested with interaction coefficients using DFT bulk calculations. Simulations using the new coefficients resulted in no clustering, indicating that the original coefficients was overestimated an not well suited for representing this system. The problems with the system getting stuck during simulation introduced by the cluster expansion method has been resolved using the second order residence time algorithm which decreased the computation time with a minimum factor of 3.

The face cubic lattice structure used in the simulation have been expanded to enable jump into octahedral site, enabling the simulation to include formation of precipitates. However, the lack of clustering using the implemented methods prevents the precipitates to form. This can most likely be solved by improving the training set or improving the way coefficients from the cluster expansion is chosen.

Sammendrag

I denne masteroppgaven er Al-Mg-Si legeringer undersøkt ved hjelp av kinetisk Monte Carlo simulasjoner styrt av energibarrierer. Arbeidet er en fortsettelse av tidligere arbeid som modellerte barrierene ved hjelp av klyngeekspansjon på aktiveringsenergier fra tetthetsfunksjonalteoriberegninger (DFT). Dette arbeidet er utvidet ved å implementere en kinetisk løst aktiveringsbarriere som sikrer at totalenergien er bevart og detaljert balanse er opprettholdt ved å inkludere totalenergien eksplisitt i uttrykket. De to metodene er begge trent på samme treningssett og simulert ved å bruke et system med $25 \times 25 \times 25$ kubisk flatesentrerte enhetsceller og legeringen består av 0.77% Si og 0.67% Mg i romtemperatur. Den nye metoden introduserte en RMSE på 38.71 meV sammenlignet med 9.14 meV for den originale metoden ved å bruke samme treningssdata. Ingen av de to metodene har tendenser til danne klynger i løpet av simuleringen men den store energidriften introdusert i det tidligere arbeidet ble korrigert av den nye metoden. En tredje barrieremodell som er observert til å danne klynger har blitt tested med nye interaksjonskoeffisienter ved bruk av DFT bulkkalkulasjoner. Simuleringer med de nye koeffisientene resulterte med ingen klynger som indikerte at de originale interaksjonskoeffisientene var overestimerte og ikke egnet til bruk i dette systemet. Problemene med at systemet satte seg fast under simulering introdusert av klyngeekspansjonen, ble løst ved hjelp av andre ordens residenstidalgoritme som reduserte utregingstiden med minste faktor 3. Den kubisk flatesentrerte gitterstrukturen brukt i simuleringen har blitt utvidet til å muligjøre hopp til oktahedriske gitterpunkter. Dette åpner mulighetene til å danne presipitater i dette systemet. Mangel på dannelsen av klynger ved hjelp av barrieremodellene forhindrer imidlirtid dannelsen av presipitater. Dette kan mest sannsynlig løses ved hjelp av å forbedre treningsdataen eller forbedre måten koeffisientene fra klyneekspansjonen er valgt.

Acknowledgements

I want to sincerely thank my supervisors Jesper Friis, Senior Research Scientists and Inga Gudem Ringdalen, Research Scientists at SINTEF Industry, Department of Materials and Nanotechnology for the impeccable guidance, trust and the endless support throughout the past year. Also a special thanks to both Jonas Frafjord, Postdoctoral Fellow at Department of Physics, NTNU and David Kleiven, PhD Student at Department of Physics at NTNU. Your guidance and irreplaceable inputs is what has shaped the work into what it is. The learning curve from working this group of people has been immense.

I also want to thank everybody who I've gotten to known during these years at NTNU for giving me the best memories during the course of these studies. I will also like to thank all the friends and family that has supported me through this.

Contents

Figures

Tables

Code Listings

Acronyms

BEP Brøndsted-Evans-Polanyi relation. 24

CE cluster expansion. 17, 26, 35, 42

DFT density functional theory. 7, 20

ECI effective cluster interaction. 18, 26

FCC face centred cubic. 7

HAADF-STEM High-angle annular dark-field scanning transmission electron microscope. 7

KMC Kinetic Monte Carlo. 9

KRA kinetically resolved activation barrier. 23, 35

MC Monte Carlo. 9

MD Molecular Dynamics. 9

NEB Nudged Elastic Band. 20

RTA residence time algorithm. xi, xii, 12, 33, 58–63

SSSS supersaturated solid solution. 6

Chapter 1

Introduction

Aluminium alloys are materials widely used by industry due to having high corrosion resistance, light weight and high strength abilities. The alloys advantage of being light-weight but strong a favourable property in the automotive industry due to cost saving and eco-friendliness [1]. With the increase of interest in lowering the CO_2 emissions, the interest of these alloys are on the rise, motivating continuous advancements in the field. One of the popular types of these alloys are the Al-Mg-Si alloys, also referred to as 6xxx alloys, which is aluminium with the addition of silicon and magnesium. The combination of these three atoms have seen to form beneficial clusters and precipitations, small coherent structures in the material creating strain effects, when exposed to ageing [2].

Ever since the introduction of the computer, numerical approaches to simulate the molecular systems has increased in importance along with computational power. Combining the numerical approaches with experimental research has proved to be a fruitful combination in order to gain important insights to the evolutions and properties of these alloys on a molecular level. Albeit the great interest, numerical models of the kinetics of the solute clustering in Al-Mg-Si alloys are still lacking in quantity, so developing these algorithms is of high relevance both for scientific and commercial purposes.

One of the numerical methods used to sample these types of alloys is the kinetic Monte Carlo method. This statistical way of doing molecular modelling samples the configurational space according to probabilities of certain moves. These probabilities are governed by transition rates, which is usually calculated by the activation energies between two adjacent states.

The method used to calculate the activation energy will therefore majorly affect the outcome of the simulation, motivating the importance of finding an expression to accurately represent the physically correct activation energies. The work done in this thesis will continue on the earlier works of Øystein Nygård during his master thesis [3] in colaboration with the SumAl project where he expanded a KMC code based on an article of Liang et al. [4] to calculate activation energies using a machine learning approach. This method is called cluster expansion, and the method did well at representing the activation energies calculated from DFT, giving it a large potential to resemble the correct physical activation energies. However, the simulations done using the new method resulted in no clustering dynamics, and a large non-physical drift in the logged change of energy. The simulation was also restricted to the FCC lattice, making it unable to form precipitates with atoms residing on other sites than on the FCC matrix, such as the favourable β'' precipitate. During simulation, the activation energies found from cluster expansion could be so small that it trapped vacancies in small energy wells, making the system stuck using CPU-time on non-productive jumps back and forth, resulting in eventually having to stop the simulation from being finished.

The work in this thesis is an attempt at finding the solutions to the experienced problems using the cluster expansion method previously implemented, in order to fully take advantage of the benefits of accurate activation energies from DFT calculations. The solutions to the problems will be split in three parts. The first part is trying to resolve the energy drift using a method that removes the dependency of the direction of the jump. This will be done using the kinetically resolved activation barrier method, a method that splits the expression for activation energy into two parts, the average activation energy and the energy difference between the two adjacent states. Using this method will need a way to calculate the total energy difference that is not too computational expensive. For this, the cluster expansion method will be attempted to find a good representation of the total energy, as well as trying a simple expression assuming contributions to the energy difference to be from nearest neighbours only. The second problem addressed will be the problem of the system getting stuck during simulations. One of the main advantages of kinetic Monte Carlo is the ability to do simulations on longer time scales. One of the goals is therefore to implement an algorithm that prevents unproductive jumps by predicting the probability for jumps that enables the system to evolve. By using the probability for unproductive and productive jumps, both the transition rates and time to escape each state will be able to be scaled to represent the physical time evolution correctly.

The third problem of the simulation being restricted to the FCC lattice will be solved by introducing the possibility for a jump to octahedral sites, which enables the system to form precipitates like β''.

The thesis will in chapter 2 introduce the theoretical background of the Al-Mg-Si alloy and the precipitation process. Furthermore, it will present the theory and demands of the kinetic Monte Carlo method and introduce the first and second order time algorithm. The different methods of calculating the activation energies used in this work is presented, which includes the method used by Liang [4], cluster expansion, kinetically resolved activation barrier and the Brøndsted-Evans-Polanyi relation.

The implementation will be presented in chapter 3, where the methods behind the new implementations will be presented. This includes a general introduction to the existing frameworks the new implementations are built on, the implemented KRA method with the use of the average barrier and the method for calculating the total energy, second order residence time and the extension allowing octahedral jumps to happen.

The results from the new implementations is presented in chapter 4. The chapter will go through the performance of the existing and new methods of calculating activation energies by benchmarking them against the activation energies from DFT calculations. The method using KRA includes the results from the cluster expansion of the total energy, which is compared to an existing method of calculating total energy for a configuration. Results for simulations using these expressions activation energy is presented and compared in terms of clustering, diffusivities and energy evolution. The results from simulations using the first and second residence time algorithm is presented and compared in terms of time spent doing an iteration, number of reversals and CPU-time spent on finishing a simulation. The chapter is closed by discussing the sources of errors, possible solutions and overall results.

The thesis ends in concluding remarks and thoughts in chapter 5 and thoughts for future improvement in chapter 6.

Chapter 2

Theory

This chapter will present the background information of the work done in this thesis, including the precipitation of the Al-Mg-Si alloy, the Kinetic Monte Carlo method and the different ways of obtaining the activation energy required to run the simulation.

2.1 Al-Mg-Si alloys

2.1.1 Hardening and precipitation

The strengthening of metal alloys is done doing by a range of different approaches according to what enhancements are needed and whether the alloy is heat treatable or non-heat treatable [2]. Typical enhancements improve properties like weight, conductivity, strength, and flexibility, resulting in more environmentally friendly, better functioning, and cheaper technology and materials [1] [5]. The different methods usually require a combination of different processes that changes the properties. The strengthening of alloys is widely used across several industries, and especially aluminium alloys have the advantage of having a high strength-to-weight ratio. The four general methods to strengthen metal alloys are by solid solution, dislocation, grain boundary, and precipitate strengthening [5]. The latter is the focus of this work and is of great interest across industries.

The aluminium alloy used for this project is under the category of 6xxx series of alloys. The 6xxx series of alloys consists of the alloying solutes silicon (Si) and magnesium (Mg), and the alloy usually consists of 0.5-1.3 weight percentage of solutes [6] [7]. Other alloying elements such as Li, Ag, Zn, Ge, Fe, Mn, and Cu can also be added to the Al6xxx alloy to increase the strengthening properties [8] [9].

The Mg and Si solutes added to the alloy will cluster and form precipitates, small cohesive structures in the material and these new structures create the interface strain between the Al matrix and the solutes[10]. The precipitation process is in the literature described as a sequence of phases [6][10]

$$\text{SSSS} \rightarrow \text{atomic clusters} \rightarrow \text{GP zones} \rightarrow \beta'' \rightarrow \beta' \rightarrow \beta$$

where the starting point is a supersaturated solid solution (SSSS). The supersaturated solutions form by heating the alloy to a temperature just below melting temperature, forming a solid solution, usually between 400 and 500°C [11]. This solid solution is rapidly cooled down in a process called quenching, which is too fast for the solution to reach equilibrium, and the solution is consequently supersaturated. The next step is exposing the SSSS to ageing which combines natural ageing at room temperature and artificial ageing at elevated temperature. The solutes out of equilibrium will start to form clusters in this process. The different ageing temperatures and times will determine the sizes and densities of the precipitates, which will determine the properties of the material. The ageing process is delicate, and over-ageing and under-ageing (too high or too low temperatures or too high or too low ageing times) can lead to both unwanted and wanted precipitation forming [12] [10] [13].

The peak strength is usually obtained after a few hours with temperatures between 150 and 200 °C. the increase in strength is caused by the increased density of the β " precipitates together with the Guinier-Preston (GP) zones. These hardening precipitates will with more ageing decrease and form more stable types [11]. The work in this project focuses on the earlier stages of the precipitation sequence, more specifically about the β" precipitates.

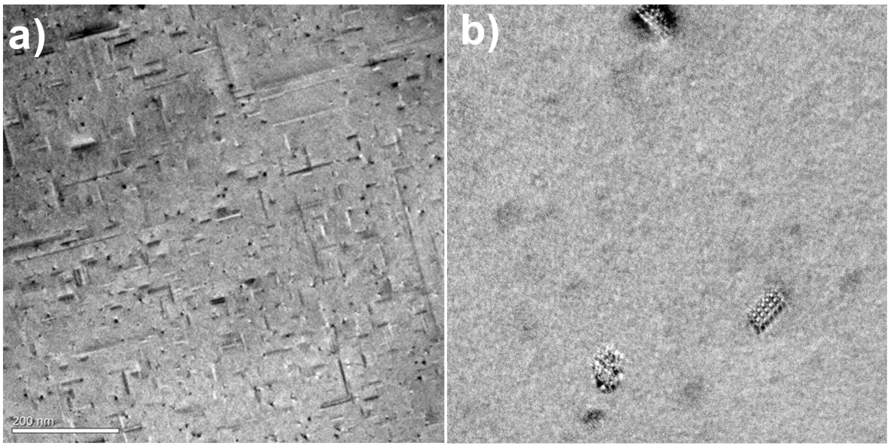


Figure 2.1: Bright field TEM image of an Al6060 alloy stored for 17 years at room temperature. a) shows the needling shape of the precipitates and b) shows a zoom on the β'' precipitates as structures made of multiple β'' eyes. Image taken by Senior researcher at SINTEF Department of Materials and Nanotechnology, Calin D. Marioara.

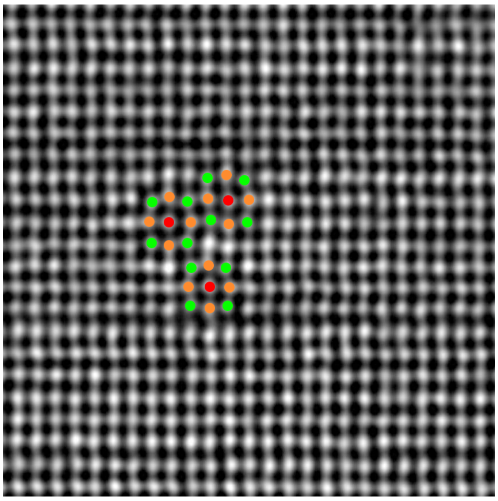


Figure 2.2: HAADF-STEM taken by Calin D. Marioara, Senior researcher at SINTEF Department of Materials and Nanotechnology. The image shows three β''-eyes breaking the symmetry of the FCC structure. The breaking of structure causes a *strain* effect.

2.1.2 Al-Mg-Si structure and β'' precipitates

Pure aluminium at room temperature has a face centred cubic (FCC) structured lattice, a close-packed lattice structure. The FCC lattice is illustrated in figure 2.3, with the corner sites illustrated as white spheres and face centred sites illustrated as red spheres. The lattice is built out of repetition unit cells, with each conventional unit cell including 4 identical lattice sites. When adding a few percentages solutes dispersed in the system, it is still assumed that the structure is similar to the Al matrix FCC. When the systems start to cluster and for precipitates, the FCC structure is no longer energetically favourable, and the crystal structure will change depending on the precipitate.

The beneficial β'' precipitation has a structure that slightly shifts the lattice cell and forms several eye-looking shapes, as seen in figure 2.2. Similar to the other metastable precipitates in Al-Mg-Si alloys, the β'' precipitates are elongated and coherent in the <100> direction and forms in needle shapes as seen in the bright field TEM image of the Al6060 alloy in figure 2.1. The cross-section of the β'' is typically between 1 and 15nm^2 and the length between 30 and 100 nm. The unit cell of this structure differs slightly from the FCC structure and is monoclinic [11]. How it forms and in what composition it is most stable is still a relevant field of study, and there are continuous updates. Compositions was for a long time believed to be Mg_5Si_6, but has been later been found to be $Mg_4Al_3Si_4$ using HAADF-STEM. However, the most stable chemical composition has calculated as $Mg_5Al_2Si_4$ using DFT. [11]. In unpublished work with density functional theory (DFT) calculations done by Inga G. Ringdalen, Researcher at SINTEF Industry, Department of Materials

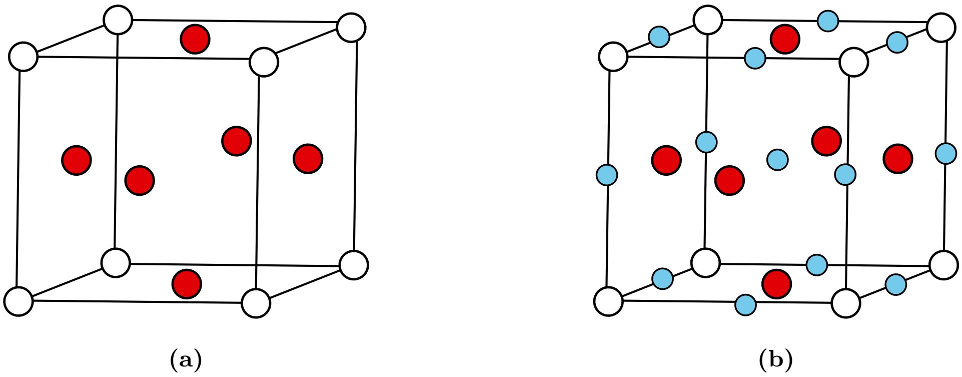


Figure 2.3: Illustration of the conventional unit cell consisting of cubic lattice sites (white) and face centred sites (red). Figure b) includes the placement of the octahedral sites (blue) in the FCC structure. The unit cell includes four lattice sites in a), with octahedral sites b) this includes 4 extra sites per unit cell.

and Nanotechnology, multiple ways of forming a β'' precipitate from an initial state were observed. One of the observed pathways is shown in figure 2.4, and the observations are a jump into a corner that creates a formation of 4 Mg that pushes the 5th middle Mg into an octahedral site in the same plane as 4 Si atoms. This Mg pushes on the surrounding Si atoms, which creates an offset, thus creating the observed eye. The octahedral site is as a shift of $(0,0,\frac{1}{2})$ relative to the atom sites in the FCC unit cell and is depicted as the blue spheres in figure 2.3b.

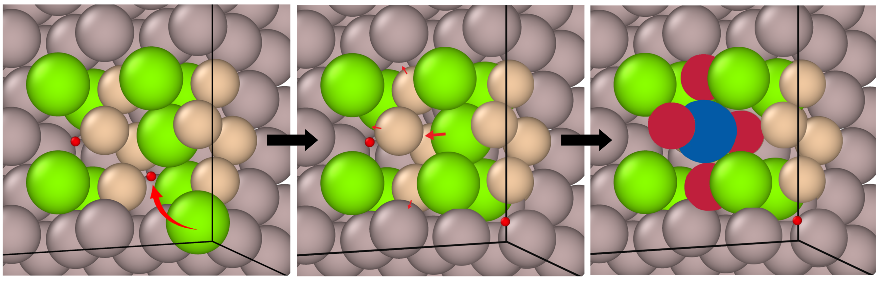


Figure 2.4: Observed steps during DFT calculations for a jump into octahedral sites creating a β''-eye First step in this event is a jump to a vacant position seen in the left picture. In this DFT calculation, the first jump pushes the Mg in the middle of four Mg into the octahederal site, creating the β''-eye seen in the last step. The layer shared by the Mg in the octahedral site and the surrounding 4 Si atoms are marked blue and red.

2.2 Kinetic Monte Carlo

This section will introduce the numerical method used to simulate the Al-Mg-Si alloys used in this project. The method used for this work is the kinetic Monte Carlo algorithm, also called the rejection-free Monte Carlo. There are several methods in order to simulate the behaviour of atomic systems, however all methods have different utilization based on their strengths and weaknesses and what is under interest.

Monte Carlo (MC) simulations have been used for various problems since first introduced by Ulam and Neumann in the '40s [14]. It is used in a wide variety of fields including in risk analysis [15], finance [16] and statistical mechanics [17]. The general idea with these methods is to use random numbers to choose moves according to their probabilities to explore a pool of possible events. The most commonly known MC algorithm is the Metropolitan Monte Carlo algorithm introduced by Metropolis et al. [18]. It is a commonly used method to study equilibrium properties of physical systems. The method samples a system by trial and error, where moves that meet a particular condition is accepted; otherwise, the move is rejected. A drawback for this method is the lack of an objective definition of time, making it unable to estimate some time-dependent properties and kinetics. Another commonly used method to simulate a system on a molecular level is Molecular Dynamics (MD) which uses the classical equations of motion to propagate the system over time. The advantage of this method is that it ensures correct dynamical evolution and can estimate time-dependent properties. However, to have accurate development of the system, the time increment has to be small ($\sim 10^{-15}$s), making it hard to simulate dynamics over longer periods of time. Both these methods are under development and have consistently improved over the years [19], making the algorithms more versatile and efficient.

In the 1960s, a new Monte Carlo method was developed to explore the dynamic evolution of an atomic system [20]. This algorithm has later gotten the name kinetic Monte Carlo (KMC) and has become a popular tool to study material properties such as surface adsorption, diffusion and growth [20].

The KMC algorithm is similar to the Metropolis MC, but it differs in KMC by performing a step at each increment and updating time for each jump to happen instead of having steps doing rejections. This makes for an efficient algorithm for systems where the escape time is significantly long, such as for systems with higher activation energies.

The steps chosen for the KMC simulation is based on rates to transition from state i to j, Γ_{ij}. These transition rates are calculated from the activation barrier E_{ij}^{act}:

$$\Gamma_{ij} = \nu \exp\left(-\frac{E_{ij}^{eact}}{k_B T}\right) \tag{2.1}$$

where ν is the attempt frequency, $k_B =$ is the Boltzmann constant, T is the temperature and E_{ij}^{eact} is defined as the energy needed to transition from a

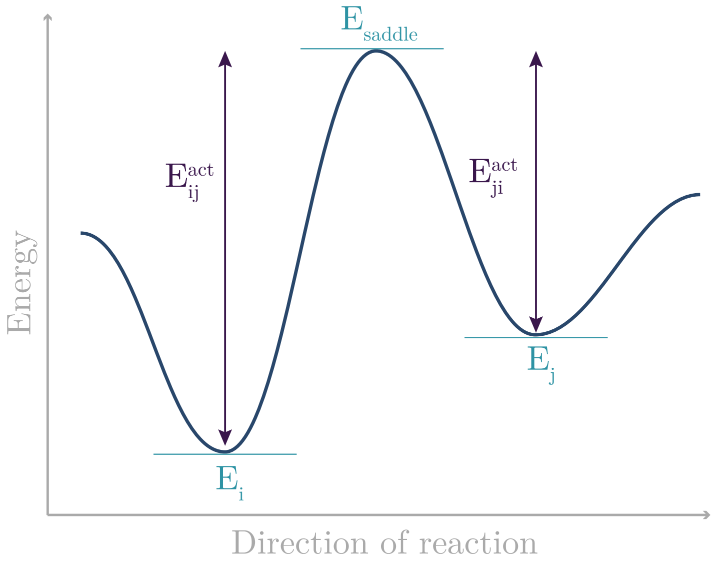


Figure 2.5: Illustration of the activation energy, which is the energy needed to perform a jump from site i to site j, $E_{ij}^{act} = E_{saddle} - E_j$

state i to a state j and is illustrated in figure 2.5. The activation energy is the energy difference between saddle and initial energy path from i to j, E_{saddle}, and the configurational energy at state i is:

$$E_{ij}^{act} = E_{saddle} - E_i \tag{2.2}$$

The calculation of activation energies will be described in section 2.3.

In order to choose an event from the N_e number of possible events, the cumulative sum of rates is used:

$$R_{i,k} = \sum_{j=1}^{k} \Gamma_{ij} \tag{2.3}$$

which is the sum of all the rates of the possible events. What events to include in this are defined by the user. For use for simulating alloys, these are usually defined as the transition between two adjacent states. In order to choose an event, a number $0 < u \leq 1$ from a random distribution is chosen, and the event that fulfils

$$R_{i,k-1} \leq uR_{i,N_e} < R_{i,k} \tag{2.4}$$

will be the event executed. Figure 2.6 illustrates an example of 7 possible events, where the event with rate Γ_2 satisfies the condition uR_7 and event 2 is chosen. Note that larger rates give a larger probability to be chosen.

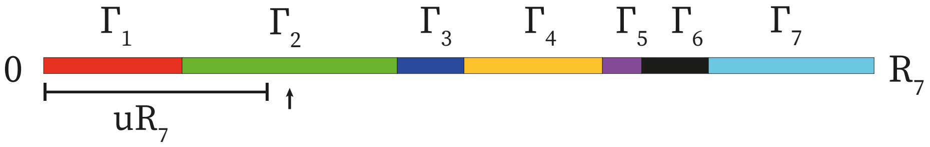


Figure 2.6: Illustration of the process of choosing which move to make next. The rates Γ_i corresponds to the rates of each possible move and $0 < u \leq 1$ is a random number. The move i chosen is the one that satisfies $\Gamma_i = uR_n$, which in the figure corresponds to the event with rate pointed to by the arrow, Γ_2.

Criteria for correct sampling

An essential requirement for the Monte Carlo of dynamic systems is to correctly describe the system towards equilibrium as $t \to \infty$. Since the method statistically samples the configurational space, the underlying conditions for reaching equilibrium must be met and the resulting sampled distribution must correspond to the distribution of the actual system. Newman and Barkema [21] describe some requirements and sum it up in three criteria: the simulation steps has to be Markovian, ergodic and obey detailed balance.

For a move from state j to i to be Markovian, the move should only be dependent on the information about the current state and the possible next state. How the system ended in its current state should not affect the next transition. Thus the probabilities to transition should only depend on state i and j, and these transition probabilities should be independent of time. The probabilities of transitioning from j to i are defined as the rate for a jump to i over the sum of all the rates R_{i,N_e}:

$$\alpha_i = \frac{\Gamma_{ij}}{\sum_{j=1}^{k} \Gamma_{ij}} = \frac{\Gamma_{ij}}{R_{i,N_e}} \tag{2.5}$$

which also satisfies the additional constraints that the sum of the probabilities must be 1:

$$\sum_{i}^{N_e} \alpha_i = 1. \tag{2.6}$$

The second criterion of ergodicity means that each state of the configuration is possible to reach within finite time. This implies that no state in the simulation cannot be visited and ensures that states are generated with their correct Boltzmann probability. The path does not need to be directly accessible from each state, but it must be able to get there if given enough steps.

The last mentioned criterion for ensuring correct sampling is the criterion of detailed balance. Detailed balance is the criterion that ensures that the equilibrium reached with $t \to \infty$ is the Boltzmann probability distribution. This is maintained by requiring that the probability of being in state j, p_j and the probability to transition to state i, α_i is the same as being in state i and transition to state j:

$$p_j \alpha_i = p_i \alpha_j \tag{2.7}$$

2.2.1 First order time algorithm

Since a jump happens at each KMC iteration, the between events must be accounted for by calculating it accordingly. Bortz et al. [22] introduced a method for calculating this time using the n-fold way algorithm, also called the first order residence time, τ. This time is equivalent to the average time the system is expected to remain in a given configuration and is given by

$$\tau = \frac{1}{R_{i,N_e}} \tag{2.8}$$

with R_{i,N_e} is the sum of all rates for N_e events. The time for the KMC system with a vacancy concentration C_v and n sites is scaled as

$$\Delta t = \frac{\tau}{nC_v} = \frac{\tau}{C}. \tag{2.9}$$

Where C is a scaling constant. The vacancy concentration can be written as an Arrhenius term [23]:

$$C_V(T) = \exp\left(-\frac{G_V^F}{k_B T}\right) \tag{2.10}$$

where the Gibbs free energy is defined as

$$G_V^F(T) = H_V^F - TS_V^F \tag{2.11}$$

where H_V^F is the formation enthalpy and S_V^F is the formation entropy for monovacancies.

2.2.2 Second order residence time algorithm

A common problem with KMC is the trapping of vacancies, also called flickering states. The flickering states occur when the energy of two or several neighbouring states are significantly lower than others, resulting in vacancy jumping back and forth between these states because it is the energetically most favourable and gives the highest rates. An illustration of this can be found in figure 2.7. These types of jumps are not very productive for a simulation as the system does not change and the residence times corresponding to the flickering usually are short. One of the methods of circumventing this is using a second order residence time algorithm (2RTA) presented in the works of Athénes et al. [24] and used in other works where flickering was a problem [25] [26] [27]. What separates this from the first order RTA is the amount of KMC-steps considered when calculating the transition probabilities. The first order uses the rates from the n_{nn} nearest neighbours, while the 2RTA also uses the n_{nn} nearest neighbours of the n_{nn} nearest neighbours to calculate these probabilities. The order of the algorithm is how many steps ahead that is considered, and higher orders come with an increased computational cost. It is often not considered higher orders than 4 [24].

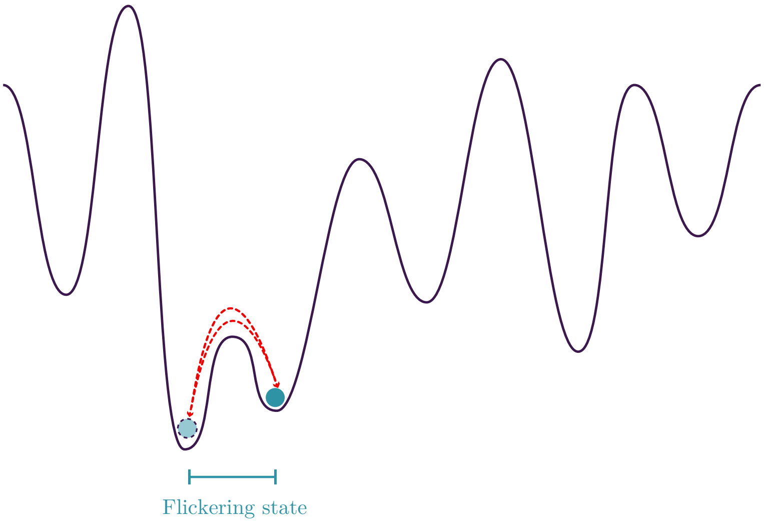


Figure 2.7: Figure illustrating the energy landscape and the flickering states trapped in lower energies.

Two important changes are introduced with the second order RTA:

1. The probability to transition must exclude the probability contribution arising from the previously counted reversals
2. Residence times needs to account for the time spent in reversals

The sites reached after jumps and reversal is shown in 2.8, where l is the start position of the vacant site, i are the nearest neighbours of l, and j is the previously visited state. k are the nearest neighbours of i apart from l.

The probability that a vacancy jumps back to its previous site from i is calculated from equation (2.5) and denoted $\hat{\alpha}_i$. The probability for a reversal jump is the probability for the jump to site i and back to site j

$$\bar{\beta}_i = \alpha_i \hat{\alpha}_i \tag{2.12}$$

The probability that the vacancy does not jump back after a jump is

$$\beta_i = \alpha_i (1 - \hat{\alpha}_i) \tag{2.13}$$

The corresponding probability that the vacancy does a reversal jump to any neighbour is

$$\bar{\beta} = \sum \bar{\beta}_i \tag{2.14}$$

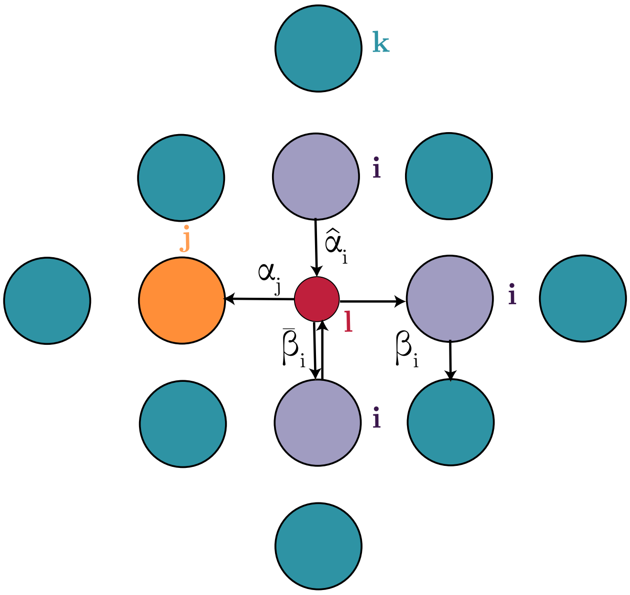


Figure 2.8: Illustration of the second order probabilities for a vacancy at site l. The neighbours possible to transition to is i and j is the last visited site and a direct jump to j is forbidden. α_i is the probability to transition to site i, $\hat{\alpha_i}$ is the probability to jump back to previous site from site i. $\beta_i = \alpha_i(1-\hat{\alpha_i})$ is the probability to escape via i and $\bar{\beta} = \alpha_i\hat{\alpha_i}$ is the probability for a reversal jump.

and the probability for a jump to any neighbour without reversal jump is

$$\beta = \sum \beta_i \tag{2.15}$$

Defining the site the vacancy jumped from as j, the probability for second order jumps (direct jumps to sites different than j or after at least one reversal) can be written as

$$1 - \alpha_j = \sum_{i \neq j} \beta_i + \left(\sum_{i \neq j} \bar{\beta_i} \right) (1 + ... + \bar{\beta^n} + ...) \left(\sum_i \beta_i \right) \tag{2.16}$$

The first term on the right hand side corresponds to the direct escaping to any neighbour $i \neq j$ and the second term is the probabilities for n reversal jumps before escape. Rautiainen et al. [25] use equation (2.16) to find the relative transition probabilities to jump back to j, P_j, and to other sites $i \neq j$, $P_{i \neq j}$:

$$\begin{aligned} P_j &= \frac{1}{1-\alpha_j} \left[\left(\sum_{i \neq j} \bar{\beta_i} \right) (1 + ... + \bar{\beta^n} + ...) \beta_j \right] \\ &= \frac{1}{1-\alpha_j} \frac{\sum_{i \neq j} \bar{\beta_i}}{\beta} \beta_j \end{aligned} \tag{2.17}$$

$$\begin{aligned} P_{i\neq j} &= \frac{1}{1-\alpha_j}\left[\beta_i\left(\sum_{i\neq j}\bar{\beta}_i\right)(1+...+\bar{\beta}^n+...)\beta_i\right] \\ &= \frac{1}{1-\alpha_j}\left[1+\frac{\sum_{i\neq j}\bar{\beta}_i}{\beta}\right]\beta_i \end{aligned} \tag{2.18}$$

which are the transitions probabilities for a second order jump.

The residence time for these jumps must take into account the reversals jumps. The time to escape, excluding back to site j, is $\sum_{i\neq j}\beta_i\tau$, with τ being the first order residence time for the vacancy at the current site from equation 2.8. The average time spent doing a single reversal is an average of all the reverse jumps to the neighbouring sites:

$$\tau_S = \frac{\sum_i \bar{\beta}_i(\tau+\tau_i)}{\sum_i \bar{\beta}_i} \tag{2.19}$$

where τ_i is the first order residence time at site i. The average time spent on the reversal after the first one is

$$\tau_S^{(j)} = \frac{\sum_{i\neq j} \bar{\beta}_i(\tau+\tau_i)}{\sum_{i\neq j} \bar{\beta}_i} \tag{2.20}$$

Second order mean residence time is thus defined as the average of the time associated with direct escapes, excluding to site j, and of the average time spent on reverse jumps prior to escaping, scaled by the probability of a second order jump:

$$\tau^{(2)} = \frac{1}{1-\alpha_j}\left[\left(\sum_{i\neq j}\beta_i\right)\tau + \left(\sum_{i\neq j}\bar{\beta}_i\right)\left(\tau_S^{(j)}+\tau+\frac{\bar{\beta}}{\beta}\tau_S\right)\right] \tag{2.21}$$

Furthermore, the residence time scaled for the system is

$$\Delta t^{(2)} = \frac{\tau^{(2)}}{C} \tag{2.22}$$

with the same C as defined in equation 2.9.

2.3 Determining the activation barrier

This section will present the different methods used by the SKMC code to calculate the activation energy E_{ij}^{act}. The original script was based on the works of Liang et al. [4] and the method of calculating the activation energy will be referred to as the Liang method. SKMC was further expanded by the previous master student Nygård [3] to be able to calculate the barrier using a Bayesian approach to Cluster Expansion based on works by Sanchez et al.

Variable	Value
ϵ_{Mg-Vac}	-0.015 eV
ϵ_{Si-Vac}	-0.025 eV
ϵ_{Mg-Si}	-0.04 eV
ϵ_{Mg-Mg}	0.04 eV
ϵ_{Si-Si}	0.03 eV
E^d_{Al}	1.29 eV
E^d_{Mg}	1.27 eV
E^d_{Si}	1.15 eV
E^d_{Vac}	0.63 eV

Table 2.1: Table showing the values from article by Liang et. al [4] used in equation (2.23).

and Mueller et al. [28][29]. The work in this project includes introducing a new method using the kinetically resolved activation barrier. This method is based on the works of Van der Ven et al. [30] and explicitly includes configurational energies. This requires a method for calculating the total energies, which is also presented.

2.3.1 Liang

A simple model for estimating the activation energy from a jump from i to j as presented in the works of Liang et al. [4] is assuming that E^{act}_{ij} depends on the bonds formed and broken between the vacancy and jumping atom when performing the jump, and what type of atom jumping. The resulting expression is

$$E^{act}_{ij} = E^d_{X_i} - E^f_{Vac} - \left(\sum_{k \in NN_j} \epsilon_{X_k Vac} + \sum_{k \in NN_i} \epsilon_{X_k X_i} \right) + \left(\sum_{k \in NN_i} \epsilon_{X_k Vac} + \sum_{k \in NN_j} \epsilon_{X_k X_i} \right) \tag{2.23}$$

where E_{X_i} is the theoretical diffusion activation energy for atom X_i jumping from site i, E^f_{vac} is the formation energy of the vacancy, and the $\epsilon_{X,Y}$ is the chemical interaction energies between the two atoms, X and Y. The two expressions in the parenthesis effectively sums the interactions between the atom and vacancy and their nearest neighbours, k, before and after they swap. The values in equation (2.23) is presented in table 2.1. The interaction with Al is zero by definition.

Calculating interaction coefficients

The method to calculate the interaction coefficients used in Liang [4] is based on the article by Hirosawa et al. [31] and the assumption is that the energy to create a bond is the difference in the two atoms, X and Y, being nearest neighbour and being infinitely far apart.

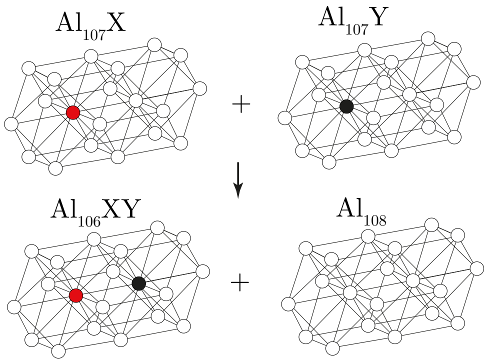


Figure 2.9: Illustration of two atoms, X and Y, in an initial position infinitely apart and a final state where they are nearest neighbours.

As figure 2.9 illustrates, the configurational energies for the atoms with no nearest neighbours, with X and Y being nearest neighbours and pure aluminium must be calculated. This can be done via DFT calculations of the bulk. The interaction coefficient can then be calculated as

$$\epsilon_{X,Y} = (E_{Al_{106},XY} + E_{Al_{108}}) - (E_{Al_{107},X} + E_{Al_{107},Y}) \tag{2.24}$$

Where 108 is the number of total atoms in the simulation cell.

2.3.2 Cluster Expansion

A method implemented in SKMC by earlier master student Øystein Nygård [3] is a local cluster expansion (CE) of E_{ij}^{act}. The method is based on the works of Sanchez et al. [28] and Mueller et al. [29] and combines a model similar to the Ising model in combination with Bayesian machine learning to find coefficients that can represent the system efficiently in the simulation. The CE method is a versatile method of linking configurations in a system to a property value, and examples of use are formation energies in binary substitutional systems [32], band gaps [33] and electronic excitations [34]. However, one of the most important uses is to obtain the total energies for Monte Carlo simulations [35] [36] [37] The method of CE of the energies takes advantage of the great benefits and increased accuracy of using first-principles calculations, but without the increased computational cost it brings.

The main idea of CE is to approximate the contributions certain configurations have on a property, in this case, the energy. This can, in principle, give an exact solution by calculating every possible configuration and including an infinite amount of clusters, but in doing so, the CE will lose its advantage of reducing the computation time. The advantage of this method is that the effect the different configurations have on a system depends quickly converges, and can be truncated accordingly. The included configurations can thus be chosen on whether they have a significant contribution to the system. These different configurations are defined as clusters and are usually defined in alloy simulations by the size, amount of sites included, geometrical shape, types of atoms that are present in each of these defined clusters. An illustration of examples of clusters is seen in figure 2.10.

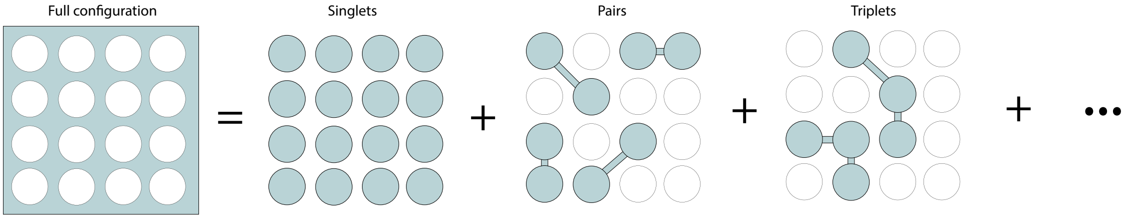


Figure 2.10: Illustration of cluster expansion using sum of contribution of clusters illustrated in this figure as singlets, pairs and triplets.

Mathematically the cluster expansion of a property consists of a set of single-site basis functions multiplied to define a basis for a type of cluster and the corresponding effective cluster interaction coefficients (ECIs) [29]:

$$F(\vec{s}) = \sum_{\vec{b}} V_{\vec{b}} \prod_{j} \Theta_{b_j,j}(s_j) \tag{2.25}$$

where the b'th basis function is defined as $\Theta_{b_j,j}(s_j)$, for site j with site variable s_j and the $V_{\vec{b}}$ is the ECI for the set of single-site basis functions. The site variable is a variable that indicates what species resides on site j. The product of the basis functions $\prod \Theta_{b_j,j}(s_j)$ is known as the cluster function and is denoted by

$$\Phi_{\vec{b}}(\vec{s}) = \prod_{j} \Theta_{b_j,j}(s_j) \tag{2.26}$$

All clusters $\vec{b}$ that are symmetrical equivalent are denoted by α, and contributes equally to the value property, and the corresponding ECIs should be equivalent. The expression for the cluster using equation (2.25) and (2.26) is thus defined for the α as:

$$F(\vec{s}) = \sum_{\alpha} V_{\alpha} \sum_{\vec{b}\in\alpha} \Phi_{\vec{b}}(\vec{s}) \tag{2.27}$$

Which is used to express both the cluster expansion for the total energies and activation energies in the following sections.

CE of the total energy

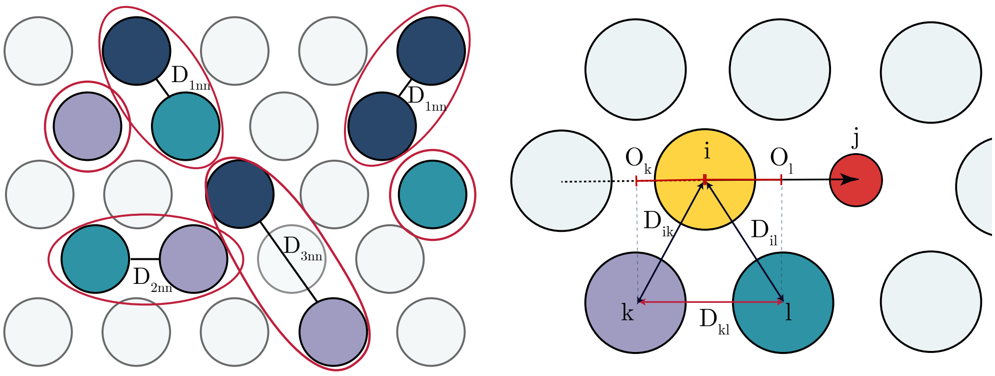
As mentioned, the CE is most frequently used to calculate the configurational energies for alloys and is also applied in the SKMC by using the KRA method presented later in section 2.4. The clusters defined for the configurational energy are usually single sites, pairs, triplets or higher order clusters, defined by the distances between sites, order, and site occupation. What order to include is chosen according to needs of precision and computation time, and a method for choosing the most fitting truncating is further explored in [38]. For use in this work, only single sites and pairs are included.

This choice simplifies (2.27) to only contain pairs, and the expression becomes the sum of the coefficients for atom type and and coefficients corresponding to all pairs.

$$E^{conf} = \sum_i \epsilon_i + \sum_i \sum_{j<i} \epsilon_{ij} \tag{2.28}$$

with ϵ_i being the single-site coefficient at site i and ϵ_{ij} the pair coefficients between site i and j which only includes the pairs within the wanted number of nearest neighbours.

The included clusters in E^{conf} is illustrated in figure 2.11 a and the distances D_{Nnn}, of Nth nearest neighbour, is the distances that defines the pair clusters. The clusters in this work are only including single sites and pairs up to the third nearest neighbour.



(a) **(b)**

Figure 2.11: Figure showing the difference between a a) cluster expansion on the configurational energy marked with red circles and b) cluster expansion the local contribution to the activation energy for a jump from i to j. CE on the configurational energy counts the clusters, illustrated as single sites and pairs with distance D_{xnn}. The clusters in the CE of the activation energy is defined by their projected distance O_k and O_l, the distances from atom k and l to the x-axes defined by the jump direction and the distances between the jump atom and the neighbouring atom D_{ik}. For the case of pair clusters, the distance for the two neighbouring atoms D_{kl} is also included.

CE of the activation energy

The earlier implemented version of cluster expansion in SKMC is for the activation energy directly and is discussed in depths in the master thesis of Nygaard [3]. This is a modified version of the CE of the configurational energy and has additional restrictions for defining the symmetrical equivalent clusters α. This is also a local CE, and the included sets of clusters are defined within a distance from the jumping atom, X_i. Since this local variable depends on what type of atom is jumping, and which direction, this is also considered when defining the cluster. Figure 2.11 illustrates the difference in defining the clusters for configurational energy and activation energy. As illustrated in 2.11b), the symmetrical equivalent clusters, α, is defined by the jumping atom i and the distance between each i and the neighbouring site, and the projected distance, O_k. The projected distance is defined as the distance from the jumping atom and the neighbour k projected on the x-axis defined by the jump direction. The increasing restrictions on symmetry increase the amount of ECIs to be considered when doing the fitting.

The activation energy for a jump from i to j is in this method given as

$$E_{ij}^{act} = V^{X_i} + \sum_{\alpha} \sum_{\vec{q}} V_{\alpha}^{\vec{q},X_i} \sum_{\vec{k} \in S_{\alpha ij}} n_{\vec{q}}(X_{\vec{k}}) \tag{2.29}$$

where V^{X_i} is an effective cluster interaction coefficient (ECI) for the atom occupying site i, $V_{\alpha}^{\vec{q},X_i}$ are the ECIs of corresponding single-site and pair clusters, $\vec{q}$ is the decoration vector, $S_{\alpha ij}$ contains vectors of the lattice site indices and $\vec{k}$ is one of these vectors. The decoration vector, $\vec{q}$ contains the atoms that defines the different clusters and contains combinations of the different clusters and additional vacancies, i.e. [Mg, Si] for a pair cluster with Mg and Si. The cluster function, $n_{\vec{q}}(X_{\vec{k}})$ is defined as

$$n_{\vec{q}}(X_{\vec{k}}) = \begin{cases} 1 & \text{for } \vec{q} = X_{\vec{q}}, \\ 0 & \text{else} \end{cases} \tag{2.30}$$

making the last sets of sums zero if the atoms residing on site k is not defined in the specific cluster. This will, in practice, sum all the ECI's of those pairs and single-sites found within the cut-off distance.

2.3.3 Finding the coefficients using DFT-data

In order to find coefficients that correctly resembles the system in the simulation, the training data must be physically correct and describe the correct dynamics. The calculations for both the total energies and activation energy for the training set in SKMC is using density functional theory (DFT), and all the calculations are done by Inga G. Ringdalen, Researcher at SINTEF Industry, Department of Materials and Nanotechnology. The DFT calculations done for this projects are using the Nudged Elastic Band (NEB) [39] and DIMER [40] method. These calculations determine the saddle points which can be used to calculate the activation energies using equation (2.2). The activation energies, paired with the configuration of the initial state, are used as input data in the training of the coefficients.

Bayesian approach

The fitting procedure in SKMC is based on the works of Mueller et al. [29] and is a Bayesian approach of linear regression. The goal of the fit is to find the set of ECI's, $\vec{V}$, that best approximate the energies $\vec{y}$ from the configurations stored in a data matrix, $\mathbf{X}$. The Bayesian approach is fundamentally built on Bayes theorem where the probability for finding a set of coefficients $\vec{V}$ given the data matrix $\mathbf{X}$ and target values $\vec{y}$ is:

$$P(\vec{V}|\mathbf{X},\vec{y}) = \frac{P(\vec{y}|\vec{V},\mathbf{X})P(\vec{V}|\mathbf{X})}{P(\vec{y}|\mathbf{X})} \tag{2.31}$$

with the vector $\vec{V}$ containing all the N_V values for the ECIs from equation (2.27), $\boldsymbol{X}$ is a matrix of size $N_s \times N_V$ that represents the number of each symmetrical equivalent cluster and $\vec{y}$ with the size of the number of structures in the data set, N_s, is the corresponding activation energy.

The denominator in (2.31) can be considered a constant because this is the probability for the energy from the training set given the structures in the training set. $P(\vec{y}|\vec{V},\mathbf{X})$ in the equation is maximized by a regular least squares regression and its intention is to increase the probability for the activation energy being correct from the training set given the ECIs and the structures from the training set. The physical insight in the system is in $P(\vec{V}|\mathbf{X})$, which is the probability for the ECIs given the structures in the system. This is also known as the *prior distribution.* It is possible to maximize $P(\vec{y}|\vec{V},\mathbf{X})$ by minimizing -ln$P(\vec{y}|\vec{V},\mathbf{X})$ so the optimal $\vec{V}_{opt}$ is chosen to be

$$\vec{V}_{opt} = \min[-\ln(P(\vec{y}|\vec{V},\mathbf{X})) - \ln(P(\vec{V}|\mathbf{X}))] \tag{2.32}$$

Assuming that the activation energies have a normal distributed $P(\vec{y}|\vec{V},\mathbf{X})$ gives

$$P(\vec{y}|\vec{V},\mathbf{X}) \propto \prod_k \exp\left(-\frac{(y_k - \vec{X}_k \cdot \vec{V})^2}{2\sigma_k^2}\right) \tag{2.33}$$

where $\vec{X}_k$ and σ_k^2 is the row k and corresponding variance of $\mathbf{X}$. Equation (2.33) is inserted into (2.32) and the first term in (2.32) can be written as

$$\sum_k \frac{(y_k - \vec{X}_k\vec{V})^2}{2\sigma_k^2} \tag{2.34}$$

The prior distribution is chosen prior to the fitting and holds information of the values of the ECIs and the values relative to other ECIs. This is achieved by choosing a prior that can be represented from a matrix **Q**, with either such that -ln$(P(\vec{V}|\mathbf{X})) = \vec{V}^T\mathbf{Q}\vec{V}/2$.

If **Q** is diagonal, the fitting will be the same as the ℓ_2 regularisation where **Q** is the regularisation matrix and the maximum likelihood estimate for $\vec{V}$ can be calculated as [29]

$$\vec{V}_{optimal} = (\boldsymbol{X}^T\mathbf{W}\mathbf{X} + \mathbf{Q})^{-1}\mathbf{X}^T\mathbf{W}\vec{y} \tag{2.35}$$

where **W** is a weighting matrix with size, $N_s \times N_s$, storing the inverse variance and **Q** is a matrix with size $N_V \times N_V$.

Using physical intuition

Choosing a good prior distribution is key for a well fitted set of coefficients. In the simplest case, **Q** is set as $\lambda\boldsymbol{I}$, where λ is a constant penalty term, and $\boldsymbol{I}$ is the identity matrix [41]. One of the advantages of knowing the properties of the dataset is the possibility to give some insights that can tune the fit accordingly. For the CE the increase of number of sites included in a cluster will decrease the contribution of that cluster, and clusters further away will contribute less than those closer. This insight can be included in the fitting procedure by including a prior that includes the relation

$$Q_{\alpha\alpha} = aD_\alpha^b + d\gamma_\alpha^e + C \tag{2.36}$$

where $Q_{\alpha\alpha}$ is the diagonal element of **Q**, γ_α is the number of sites that defines the cluster, and D_α is the average distance between all the sites defined in the cluster, i.e. the jumping atom and the corresponding single sites or pairs. The hyperparameters, a, b, d, e and C from equation 2.36, are not given and are fitted with the data in the fitting procedure. However, an important requirement in order to reduce $P(\vec{V}|\mathbf{X})$, is the requirement that the hyperparameter values are all larger or equal to 0.

Scoring the fit

An important factor for determining coefficients is not only how small the errors of the training data is, but also how well the model performs on new datasets. Decreasing terms like root mean square error (RMSE) will do well on the training set, however it does not take into account the possibility that the model might be overfitted. An overfitted model is not flexible to new configurations and can over– or underestimate contributions configurations have on the system. A way to measure how well the model performs on a new system is called cross validation [42]. The method splits the training data into the training set used for calculating ECI's, and into a validation set that tests the performance of the fitted coefficients. The RMSE of the validation sets are called the CV score, and lower CV score is indicating a good fit.

The k-fold cross validation splits the dataset into k partitions, where one of the partitions are used as validation set and the rest is used as training data. All the k partitions will be tested as the validation set and the average of the RMSE for these sets will be given as the CV score.

Another method used for fits with high coefficient-to-data ratio, is to partition the training set in to N_d partitions, where N_d is the number of entries in the dataset. This is called the leave-one-out cross-validation (LOOCV). This tests all elements of the dataset against the fit and has larger computational cost than the k-fold cross validation.

2.4 KRA-method

The activation energies calculated with the previous methods are only dependent on the local configurations and the ECI's does not consider the configurational energies E_i and E_j explicitly. The configurational energies are just implicitly included in the expression for the barrier (2.2), and any errors in fitting these ECI's are directly going to impact the system's energy evolution and not take in to consideration the underlying thermodynamic processes that drive the system to equilibrium. This breaks the demands of detailed balance because the distribution is slightly shifted. Consequently, the energy evolution will be dependent on the path that the vacancy chooses, and the end result may be a system with an non-physical final state. A solution to try to circumvent this is presented in the works of Ven et al. [30], and is used in later works using different studies [43][44]. The KRA method introduces the splitting of the expression for the activation energy (2.2) into two parts: the kinetically resolved activation energy (KRA), Q and the difference in total energy $\Delta E = E_j - E_i$

From [30] it is given that

$$Q = E_{saddle} - \frac{1}{n}\sum_{k=1}^{n} E_k \tag{2.37}$$

where k is one reachable endpoint out of n points from E_{saddle}. For SKMC, the number of reachable endpoints are 2, i and j, thus

$$Q = E_{saddle} - \frac{1}{2}\left(E_j + E_i\right) = \frac{E_{ij}^{act} + E_{ji}^{act}}{2} \tag{2.38}$$

Figure 2.12 shows that Q can be interpreted as the average activation energy for the forward and backward jump and is independent of the direction of the jump.

After finding Q and ΔE the forward activation energy can be calculated as

$$E^{act} = Q + \frac{1}{2}\left(E_j + E_i\right) - E_i = Q + \frac{1}{2}\left(\Delta E\right) \tag{2.39}$$

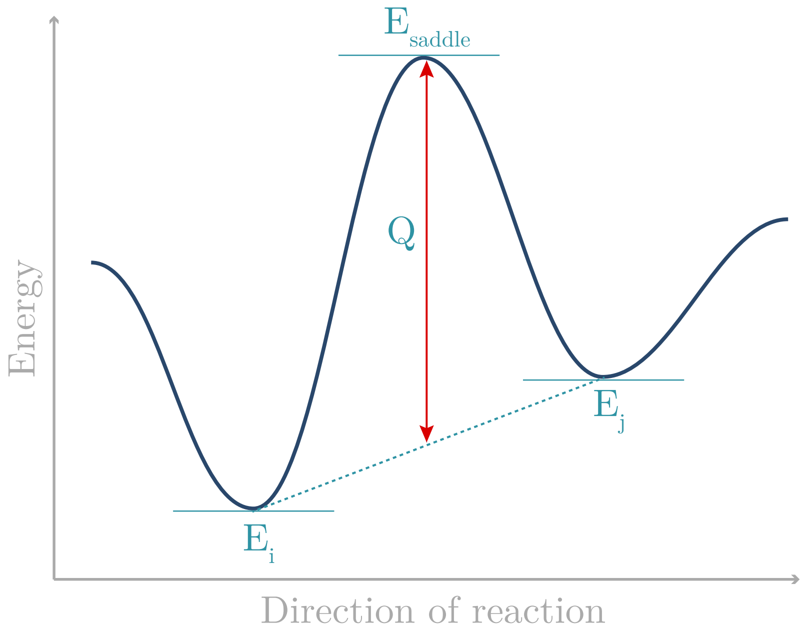


Figure 2.12: Illustration of the kinetically resolved activation energy barrier with the average activation energy for the jump from i to j and back being corrected by the total energy difference $\Delta E = E_j - E_i$.

Q can be calculated as a constant for the type of atom jumping, or it can be approximated depending on local configurations using methods like cluster expansion. For this work, the latter is chosen. The method used for the average is the same as the cluster expansion for the activation energy (2.27) and is fitted using the same Bayesian approach. However, instead of ECI's, the coefficients for each symmetrical cluster α is renamed *kinetic effective cluster interactions* (KECI's), K_α, representing the effect a pair or single-site has on the average barrier. The fitting is done with training data that pairs the forward and backward energies with an initial configuration and is fitted with y being the average of the two energies.

The coefficients from the CE of the total energies are used to calculate the total energy difference. The simulation of the system would be significantly slowed down if the configurational energy for the entire system would have to be calculated for each step. In the simulation only ΔE is needed, and since the only change between two states of the system is due to the jump, the only calculations needed are clusters that include the jump from i and the vacancy at j.

2.5 BEP-relation

Another simple method to approximate the activation barrier is using the Brøndsted-Evans-Polanyi (BEP) relation [45] [46]. This relation is assuming that the activation barrier is linearly dependent on the configurational energy, $\Delta E = E_j - E_i$ and a constant E_0 dependent on the type of jump atom

$$E_{actij} = E_0 + k\Delta E \tag{2.40}$$

where k is a constant. The BEP model is related to the KRA method in section 2.4 if Q is fixed and the constant $k = 1/2$. Physically the BEP relation proposes that the activation energy is not dependent on the transitional state but the energy of the two end configurations.

Chapter 3

Implementation

This chapter will present the implementations of the methods from chapter 2 which is used to obtain the results for this work. Most of the work done in this thesis has been the implementation of new methods and expanding the existing frameworks of the SKMC. The summary of the KMC algorithm including the new implementations is illustrated in figure 3.1. The existing KMC code is work done by Øystein Nygård previous master student at Department of Physics, NTNU and Jesper Friis Senior Research Scientist at SINTEF Industry, Department of Materials and Nanotechnology. The frameworks and implementation of observers and cluster expansion is further documented in the Appendix B of [3].

3.1 Existing frameworks

This section will briefly describe the existing frameworks for the SKMC, which is the fundamental code the new implementations done in this work is built on. The earlier implemented frameworks of SKMC is made for easily being expanded to new functionalities and logging parameters. The detailed version of the documentation can be found in Appendix B in the works of Nygård [3].

The source code of the KMC simulation is written in C due to the advantage of it being computationally fast. The SKMC framework has an Python interface to be able to start the simulation and to do the CE described in section 2.3.2. The frameworks of the simulation are built using C structs, enabling to save the lattice and parameters in memory to be accessible to the program throughout the simulation. Python can access these structures by using Simplified Wrapper and Interface Generator (SWIG)[1], which expands the struct to act as Python classes. The Python interface is made user friendly with most parameters changeable using flags on the commando line. To be able to modify the program with different events and methods of calculating the activation energy, the program is built around the `event_generator` struct. The

[1] Documentation and download can be found at swig.org

event_generator struct defines the method of calculating the activation energy, rate, performing the jump and the event_descriptor struct. This adds the possibility to use multiple types of events and calculators in the same simulation. The event_descriptor struct is used to check if a jump to a certain site fulfils the requirement for this type of event.

The existing choices of calculating the activation energies are the Liang method from the Liang article [4] and CE based on Sanchez et al. [28] and Mueller et al. [29] described in section 2.3 and is implemented by Nygård [3]. In order to use CE method in the simulation, a json file with the ECIs must be provided. This json file is an output file from the fitting process described in section 2.3.2, which is implemented in Python and C by using the same lattice framework as for the simulation.

The fitting of the ECIs is done on a dataset made from POSCAR files, a file format from the Vienna Ab inito Simulation Package (VASP). The POSCAR files needs to be a configuration with the jump atom and the vacant position provided with the corresponding activation energy for the jump included as a comment. The POSCAR format is read using the Python package Atomic Simulation Environment (ASE)[2] and is therefore expandable to support files that ASE is supporting.

The ECIs are saved in a struct defined as Eact_data, which is used when calculating the contributions that the local configuration has on the system. This is further described in the Nygårds thesis [3].

The system and parameters are logged using an observer struct that logs in user set intervals. The observer struct enables all output files to be logged in a systematic way and is initialized at KMC start.

3.2 KRA method

The KRA method from section 2.4 is briefly described in this section. This is a new implementation of this work and is initialized from the Python interface. The implementation is built on the frameworks of the CE code, since the Q from equation (2.38) is chosen to be based on the same configuration and data as for the CE of the forward activation energy.

3.2.1 Cluster expansion of Q

As mentioned, the CE used for Q in the KRA method can be defined using the same struture as the activation energies. The new implementation gives the user the choice of using the activation energy or the average, Q when creating the input file used for the fitting procedure. The option using the average needs the activation energy for the backward jump to be added to the POSCAR files on the form 'DEnergy_rev=XX.xxxx'. The KECIs resulting from the fitting

[2] ASE Package for setting up, manipulating, running and visualizing atomistic simulation. More info found at https://wiki.fysik.dtu.dk/ase/

process are similarly to the ECIs from CE of the forward barrier, saved in a json file that is used during simulation.

3.2.2 Cluster expansion for total energies

The cluster expansion for the total energies is implemented to work for the same POSCAR files from the training set by using the ASE package and by including the total energies for the configuration in the POSCAR file.

The configurations in the POSCAR files are initialized as lattice configurations with periodic boundary conditions. The use of periodic boundary conditions ensures that every site has the correct number of neighbouring atoms. In the current implementation the single-sites and pairs are defined. The number of each atom type is counted and defines the single-sites. Types of pairs are pre-defined by saving the pair distance in an array. Combinations that are equal such as [Mg, Si] and [Si, Mg], are considered the same pair. The counted configurations are stored using a dataframe using the Python package pandas[3]. The dataframe is saved to a CSV file with the columns representing type of cluster and the row corresponding to a data entry which is used as input for the fitting process.

The fitting process is done in a separate script using the in-house regression program used for the activation energies. This is possible by reusing only the regression part of the earlier implemented code, as this is only dependent on a input matrix, $\mathbf{X}$ and a set of target values $\vec{y}$. Reusing this code also opens the opportunities of including a defined prior function as discussed in section 2.3.3.

3.2.3 KRA in the simulation

The KRA method is implemented by using the existing framework of `event_generator` . Since this method is similar to the CE method of the activation energy, many of the functions are reused for this method. Due to the fit of the Q being defined using the same system as the direct activation energy, calculating Q is used using the same struct for saving data, `Eact_data` and initialized from a json file storing the KECIs.

The ECIs for the total energy difference are stored to memory using a similar struct as `Eact_data` , `KRA_CE_data` defined in listing 3.2. When calculating the energy difference between the state before and after a jump, only the included nearest neighbours of the vacancy and jump atom are affected, leaving rest of the system is unchanged. This is taken advantage of in the implementation by only calculating the differences in types of neighbour pairs before and after the jump. Since the number of each atom type is unchanged during the simulation and thus has no effect on the total energy difference, this is not included in the `KRA_CE_data` struct.

[3] Open source library prividing tools for data structures and data analysis. More info at https://pandas.pydata.org/

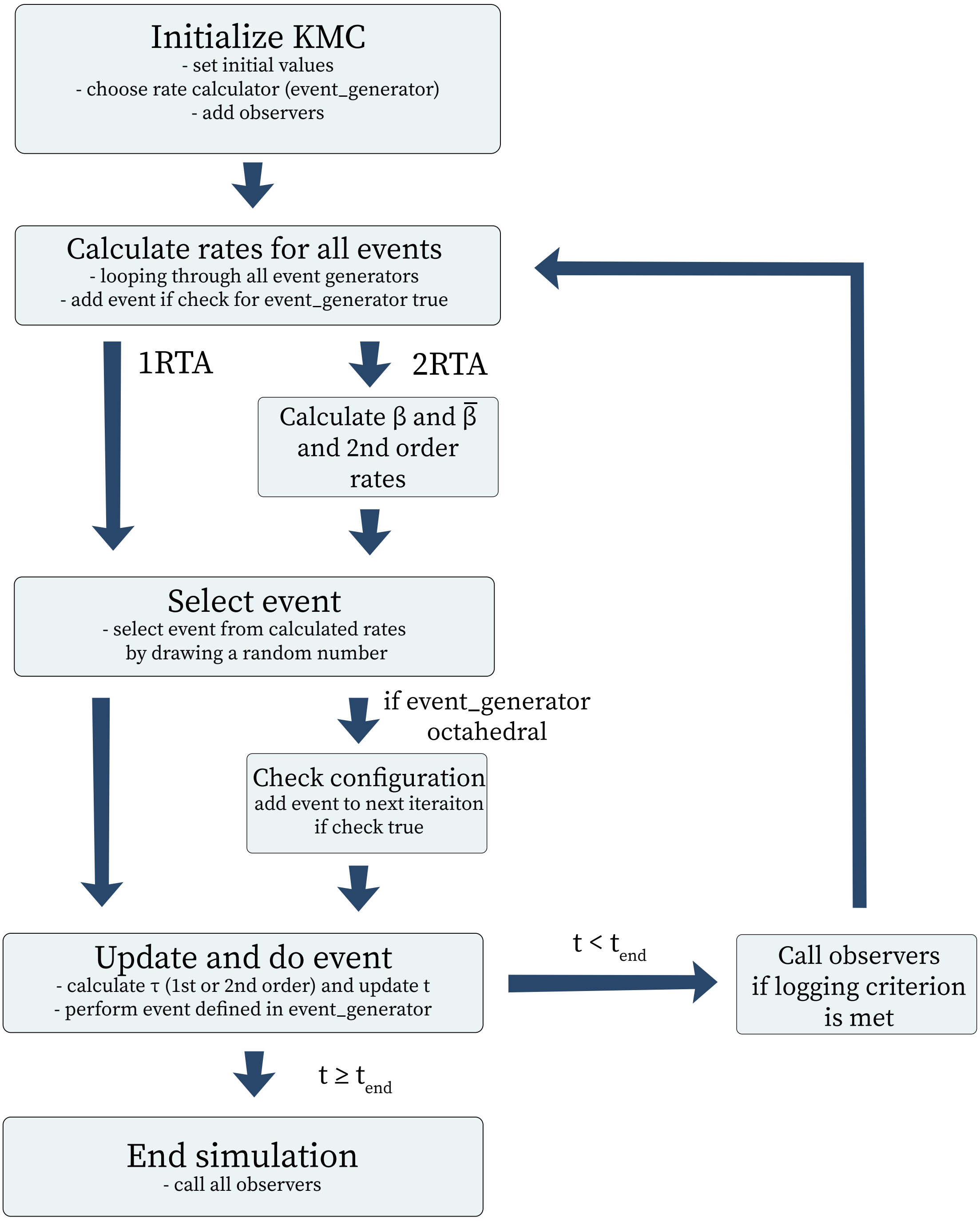

Initialize KMC
- set initial values
- choose rate calculator (event_generator)
- add observers
Calculate rates for all events
- looping through all event generators
- add event if check for event_generator true
1RTA
2RTA
Calculate β and $\overline{\beta}$ and 2nd order rates
Select event
- select event from calculated rates by drawing a random number
if event_generator octahedral
Check configuration
add event to next iteraiton if check true
Update and do event
- calculate τ (1st or 2nd order) and update t
- perform event defined in event_generator
$t < t_{end}$
Call observers if logging criterion is met
$t \geq t_{end}$
End simulation
- call all observers

Figure 3.1: Flow chart showing the KMC program with the addition of the new implementations.

```

  /** Struct for storing the ECI coefficients for the cluster expansion for total
   energies used calculate the total energy difference.
       . */
typedef struct{

  int n_elem; /**< Number of different elements */

  double **coeffs; /* 2D array storing 'n_elem'x'n_elem' coefficients */

} pair_coeffs;


/** Struct for storing the ECI coefficients for the cluster expansion for total
   energies used calculate the total energy difference.
       . */
```

Code listing 3.1: The struct used to save the data for pair coefficients.

```

typedef struct _KRA_CE_data {

  int n_elem; /**< Number of different elements in the lattice, excluding the
    vacancy. */

  int n_nn; /**< Number of nearest neighbour included (1NN 2NN 3NN etc) */

  pair_coeffs *pairs; /* Pointer coeffs that points to a list of 'n_nn' matrices of size
                        'n_comb'x'n_comb' containing the coefficients. */

} KRA_CE_data;
```

Code listing 3.2: The struct used to save the CE data for calculating the total energy difference using the KRA method.

3.3 RTA

This section will present the implementation of the algorithm in section **??** used to avoid the direct reversals. The implementation of this algorithm is not changing the physics of the system, but is enabling for a more efficient simulation.

The RTA is implemented so the user can choose to use which order RTA to use based on how much time the system is expected to spend on reversals.

This is done by initializing the RTA in the struct seen in listing 3.3. The `RTA` struct saves what function to use when choosing the event and calculating the residence time. For the second order, the function for choosing event also calculates the β, $\bar{\beta}$ and the second order rates from section 2.2.2. These calculations are saved in the `second_order_data` struct presented in listing 3.4 in order to be used in calculating the residence time after the event is chosen.

The logging of RTA is done by using the `observer` struct and logs the number of jumps, jumps back to previous site and CPUtime spent during simulation.

```

struct RTA{
  int order; /* Integer to indicate which order residence time */
  int (*choose_event)(KMC *kmc, double *R, double u);/**< Function pointer to the
                       function that chooses the event and returns the index
                       of the event in the kmc->events list*/
  double (*get_time)(KMC *kmc, double sum_rates);
                       /**< Function pointer to the funciton that calculates the
                         residence time. */
  second_order_data *data; /**< Pointer to the data needed to calculate residence time */
};
```

Code listing 3.3: RTA implementation in KMC

```

 /** Struct holding all information needed to do a second order jump and
   * and calculate the second order residence time */
typedef struct {
   double sum_beta_bar; /**< Sum of all 'beta_bar' for each event */
   double sum_beta_bar_no_j;/**< Sum of all 'beta_bar' except for last visited
                                   site. */
   double sum_beta; /**< Sum of all 'beta' for every event */
   double sum_beta_no_j; /**< Sum of all 'beta' except for last visited
                                   site. */
   double alpha_j; /**< The probability for a jump back to previous visited site
                              */
 } second_order_data;
```

Code listing 3.4: Struct saving the β and $\bar{\beta}$ values from section 2.2.2 used to calculate the second order rates and second order residence time

3.4 Jump to octahedral sites

The implementation that enables jump into octahedral sites is described in this section, which is the first step to correctly simulate the creation of the β'' precipitate. In order to be able to create the β'' in the simulation process, the program has to be adapted to include the possibility for a jump into the octahedral sites discussed in section 2.1.2. The inclusion of octahedral sites needs to be included in the lattice from the start, and this is implemented by including this as an site, consequently doubling the number of sites in the system. To differentiate the octahedral site from the FCC sites, each site is assigned a site tag in the form of an integer. This enables the program to know what type of site the atom or vacancy when performing events or calculating activation energies.

The octahedral event is built around the `event_generator` struct, enabling the customization needed to calculate the rate and perform these types of events.

Since the neighbourlist is calculated from the nearest sites on the lattice based on distance, the octahedral sites would double the size of the neighbourlist. The code was therefore implemented to have customizable cut-offs for the number included octahedrals, thus occupying less memory and reducing the time spent on looping though the neighbourlists.

3.4.1 Requirements and defining rates for jumps to octahedral sites

To find if an octahedral event is to be added, the system is implemented to check if there are any neighbouring octahedral sites that fulfills the pre-defined requirements of a jump. The requirements currently implemented are defined from observed jumps in DFT calculations discussed in section 2.1.2. This implementation checks for 4 Si in the same layer surrounding the octahedral sites, and and if the two remaining sites consists of one vacancy and one Mg atom. If this is true, the event will be added to the next iteration as a possible event. The reason for adding the even here and not when adding the other events to FCC sites is to save computation time. Since the event is defined to happen only at correct configurations, the activation energies are only necessary to implement using The current implementation does not calculate the rate using the activation energies, but this is easily implemented using CE or other methods to calculate the activation energy for the octahedral event.

Chapter 4

Results and discussion

This chapter will present and discuss the results for this work on the SKMC project, and will compare the new implementations to the work done previously in SKMC by last years master student Øystein Nygård [3]. The work is mainly divided into three parts: the three methods of calculating the activation energies, the ability to jump into octahedral site, and the second order RTA. The activation energies will be compared in two parts. First part will go through the results from the fitting process and each energy will be benchmarked with the DFT calculations. The resulting activation energies will be used to simulate the system using equal initial configurations. These simulations will be compared in terms of forming of clusters, diffusitivities and energy evolution.

The results from using enabling a jump to octahedral will be shortly presented followed by the results from implemented the first and second order RTA method.

4.1 The training set

All the data for energies used for this work is made from calculations done by Inga G. Rindalen, Research Scientist at SINTEF Industry, Department of Materials and Nanotechnology, on a $3 \times 3 \times 3$ system of unit cells, consisting of 108 atoms of different configurations of Al, Si and Mg, with one or two vacancies. The supercell has periodic boundary conditions and a lattice parameter $a = 4.0396$. The calculations are a combinations of DFT calculations using the The Vienna Ab initio Simulation Package (VASP)[1], and using NEB [39] and DIMER [40] for finding the emigration path.

DFT calculations are performed using the PBE-GGA functional [47] with a plane wave energy cut-off of 400 eV. Force convergence was set to 0.01 eV during ionic relaxation, convergence criterion for the total energy was set to 10^-5 and the gamma sampling of 0.08 k-point per Å was used to model the

[1] For more information see https://www.vasp.at/

Brillouin zone.

To be able to be used by the cluster expansion method without the atom seeing itself as its own neighbour, the dataset it was padded to be a super cell of 4 × 4 × 4 unit cells. The resulting dataset is configuration of 256 atoms, with the extra added atoms being Al. The dataset used in the previous works of master student Nygård [3] consisted of 480 different jumps and did not include an extra vacancy. For this project the dataset was expanded to 877 different jumps total, whereas 67 was removed for having more than 35 solutes or being jump to octahedral. The jumps to octahedral was removed as these was overrepresented in the training set due to testing, and because the jumps into octahedral sites are not represented by the FCC lattice system used in the fitting process. The new datasets consists of larger variety of configurations compared to the dataset used in earlier works[3] and includes configurations with one extra vacancy in the system.

The dataset is included with the activation energy for the forward jump from atom site i to vacant site j, the activation energy for the corresponding reverse jump and the total energy for the configuration. For the calculations of the total energy the unpadded version of the dataset was used, as the inclusion of Al is not represented in the calculated total energy.

The number of solutes in the dataset is shown in table 4.1, which shows the range of configurations represented by the dataset. The number of datasets from 0-3 is represented the most by this dataset and larger

# Solutes	# Data entries
0-3	586
4-7	97
9-11	77
12-15	24
16-19	21
20-27	12

Table 4.1: Table showing number of solutes present in the dataset. Number of solutes in the left column is the number of Mg and Si and right column is number of data points containing the number of solutes.

4.2 Activation energy

This section will present the results from the calculation of the different activation energies. All the methods except Liang [4] method is based on machine learning principles of cluster expansion discussed in section 2.3.2. The term 'cluster' will in this section indicate the different single-site and pair configurations illustrated in figure 2.11 and is not to be confused with clusters formed during the simulation process. Both CE and KRA uses the implemented version of CE, and earlier results from the CE method with an older dataset without the extra vacancy can be seen in the master thesis of Nygård [3]. The current CE and KRA is both done using the expanded training set presented in previous section.

4.2.1 Forward barrier

This subsection will present the CE of the forward activation energy, E_{ij}^{act}, with the new dataset. The CE is done using the method presented in section 2.3.2 and using the implemented CE method in the SKMC code.

The range of values of forward activation energies for the dataset are plotted in figure 4.1, with blue corresponding to jump atom being Si, orange to Mg and green to Al. The activation energies, E_{ij}^{act} has a variation of activation energies with values ranging from 0.1 for the lowest values, to 0.9 for the largest values. E_{ij}^{act} for jumping Al is on average larger than E_{ij}^{act} for Si, which is in turn is on average larger than E_{ij}^{act} for Mg.

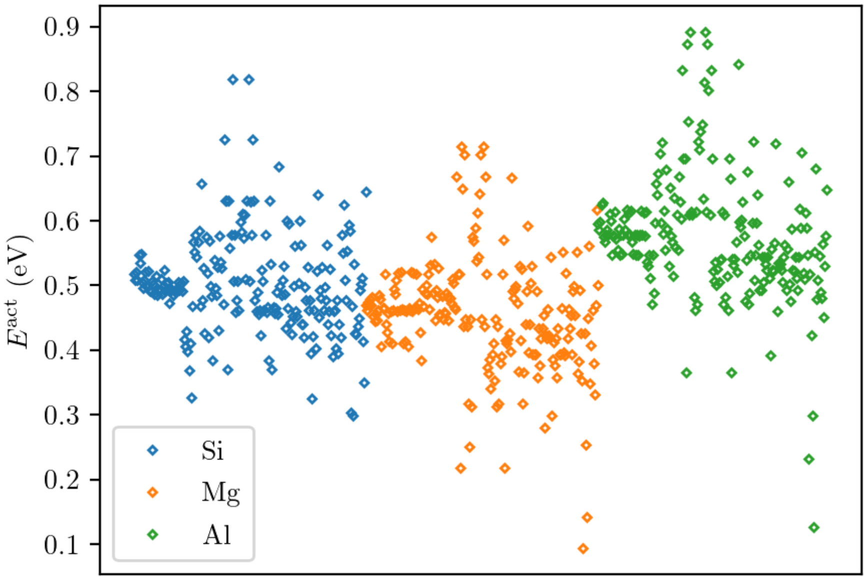


Figure 4.1: Plot of activation energies from DFT calculations for jumping atom Al, Mg and Si.

The included clusters defined for the activation energy are single sites presented in the leftmost table in table 4.2 and pairs who are a combination of two of these single-sites with pair distance defined in the rightmost table in table 4.2. This results in 11 distance and offset defined single-site clusters and 66 defined pair clusters. Adding the 3 different possibilities for jump atoms, Al, Mg and Si, and the 3 combinations of atoms and vacancies positioning the defined sites, the number of total possible ECIs are 1767. This also include the case of a jump in pure aluminium.

Distance from X_i	Offset
2.856	-2.856
2.856	-1.428
2.856	0.0
2.856	1.428
4.040	-2.856
4.040	0.0
4.040	2.856
4.947	4.285
4.947	2.864
5.713	5.713
6.387	5.713

Pair-distance
2.856
4.050
4.960
5.728

Table 4.2: Table presenting the distances from the atom at site i, X_i, and offsets described in figure 2.11b, and distance between the pairs used to define the clusters.

During the fitting process, only the ECIs corresponding to clusters found in the dataset was used. For the current dataset, 658 different clusters were used, and the rest was removed before doing the fitting procedure. Of the 810 data entries, 154 was removed for having the same ECI combinations and energy 0.5 meV within each other. The resulting data matrix $\mathbf{X}$ was of shape 658×656 representing the number of ECIs and the number of data entries. The corresponding activation energies is saved in an array as target values, $\vec{y}$, of length 656.

The fitting of the hyperparameters presented in equation (2.36) was done by trying 10 000 combinations of different input parameters. The attempted values for the power values b and e were integers 0 to 6. For the prefactor for the distance, a, 100 logarithmically spaced values between 10^{-20} and 10^{-17} were used. For prefactor d, 100 logarithmically spaced values between 10^{-7} and 10^{1} was used. The constant factor, c, had 100 logarithmically spaced values between 10^{-20} and 10^{-18}. The scoring method used was LOOCV due to the large coefficient-to-data ratio. The trial values was decided from trying a larger range of values and from the results from previously fitted hyperparameters [3].

The chosen fit with the CV score and RMSE is presented in table 4.3, along with the hyperparameters for the fit. The predicted values are benchmarked against the DFT calculations in figure 4.2 and the values falling on the line indicates a good fit. The RMSE is higher than reported using the previous dataset which was 1.77 meV [3]. The dataset used previously does however not have the possibility for an extra vacancy, decreasing the number of possible ECIs to 822, and thus decreasing the coefficient-to-data ratio.

Value	CV	RMSE	Hyperparameters
E_{ij}^{act}	27.14 meV	9.15 meV	$1.172 \cdot 10^{-18} D_{\alpha}^{1} + 0.2212 \gamma_{\alpha}^{2} + 2.807 \cdot 10^{-19}$

Table 4.3: Table presenting the chosen fit using LOOCV score and RMSE. The values of the prior function are the hyperparameters defined in equation (2.36).

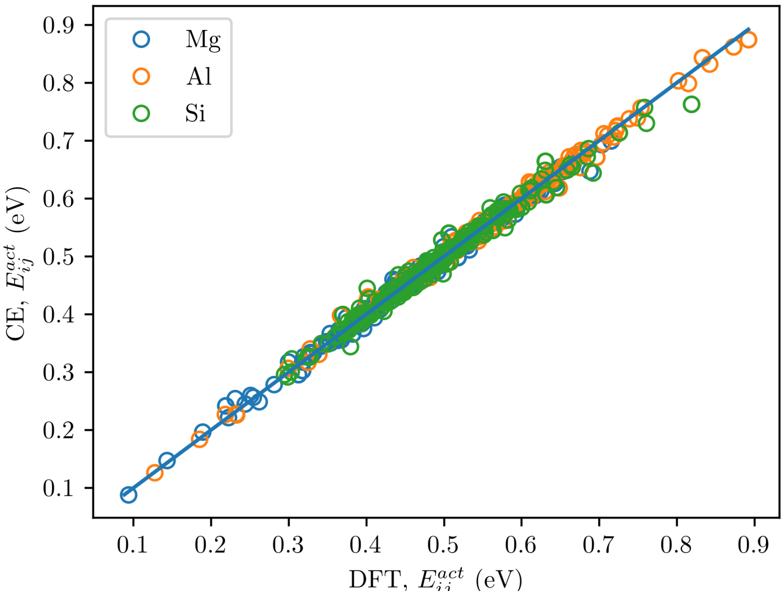


Figure 4.2: Fit of the activation barrier using CE with the predicted E_{ij}^{act} on the y-axis and E_{ij}^{act} calculated from DFT on the x-axis. The jump atom at site i are marked with different colors.

The value of hypermeters decided from the fit determines how much the distance and the order of the clusters impacts the coefficients. The prefactor, d and the power, e of the order, $\gamma_{\alpha} = \{1, 2\}$, causes the prior function to adjust the coefficients down as the order increases. The prefactor of the distance D_{α} is significantly lower and will thus have a smaller impact on the resulting coefficients than the order.

The fitted ECIs, $V_{\alpha}^{\vec{q}X_i}$, from equation (2.29) and from the fit presented in table 4.3 are plotted in figure 4.3 and shows the single-sites plotted to the left and pairs being 1st, 2nd, 3rd and 4th nearest neighbours. The distance in nearest neighbours from the jump atom is denoted by the different symbols from 1st to 4th nearest neighbours. The absolute value is seen to decrease for increased distance to the jumping atom. It is also seen to decrease as the distance between the pairs increases. The largest contributions to the activation energies are nearest neighbours single-sites with values ranging from 0.13 eV for the ECI for Al4Si to -0.1eV for the ECI for Mg4Si. The number 4 indicates that both of the ECIs is for the same site relative to the jump atom. The sites defined in these two clusters are the nearest neighbour of both the vacancy and the jump atom, indicating that this site has the largest impact on the activation energy. The type of atom interaction has different effects on the activation energy. The effect on the activation energy is positive ECIs contribute to increasing the barrier, creating a less likely jump. This means that clusters like Al4Si, a jump for Al with Si as nearest neighbour, decreases the probability for this jump to happen. The negative coefficients contribute to decrease of the activation energy, increasing the probability for the jump to happen, which is the case for the Mg4Si, meaning a Mg close to Si has an increased probability to be chosen. The ECIs corresponding to a jump in pure aluminium, V^{X_i} from equation (2.29), is presented in table 4.4 and is compared to the values calculated by DFT. These ECIs are the main contributor to the activation energy with values $\sim$ 5 times the largest ECI for the single-site cluster.

The truncating of the cluster expansion is seen to be reasonable as the ECIs significantly decreases with the order of the clusters, but also the distance from the jump atom. This indicates that the activation energy is a local dependent variable and including sites further away would not contribute much to the activation energy.

Atom	ECI	DFT
Al	595.470 meV	594.860 meV
Mg	467.646 meV	470.070 meV
Si	515.611 meV	517.880 meV

Table 4.4: Table with the fitted ECIs corresponding to a jump of each atom in pure aluminium compared to DFT calculations.

4.2.2 Liang

The activation energies calculated using equation (2.23) was benchmarked with the DFT calculations of E_{ij}^{act} and is presented in figure 4.4 a. The interaction coefficients used in the Liang [4] method were recalculated with the bulk calculations calculated from DFT using equation (2.24). The resulting coefficients are given in 4.5 the calculated E_{ij}^{act} using the new coefficient is plotted in 4.4

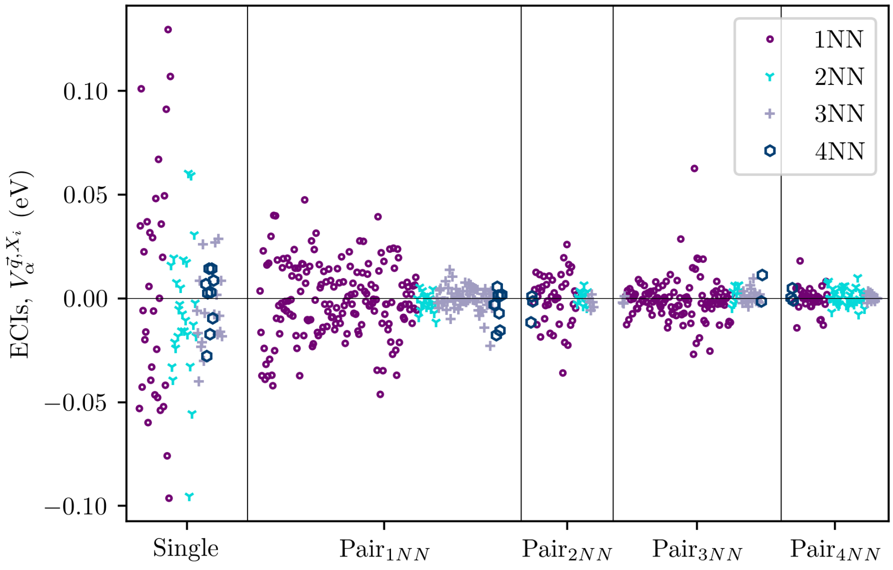


Figure 4.3: Plot showing the fitted ECIs for the different defined clusters. It is separated in single-site contributions and pair contribution with xNN indicating the pair being xth nearest neighbour of each other. Different symbols marks the distance from the jumping atom.

b. The absolute values of the interaction coefficients calculated from DFT are for the interaction between atoms seen to be lower than the Liang coefficients. The physical interpretation of this is the interaction coefficients between the Mg-Si is seen to be more attractive and Si-Si and Mg-Mg to be more repulsive than the calculated coefficients using DFT bulk calculation of 108 atoms.

Similarly to the coefficients from Liang, the activation energies calculated from DFT is not well described using this method. The range of the activation energy calculated using the coefficients from DFT is seen to have a wider range of activation energies than the coefficients from Liang. However, none of them is seen to have as large range as the DFT calculations.

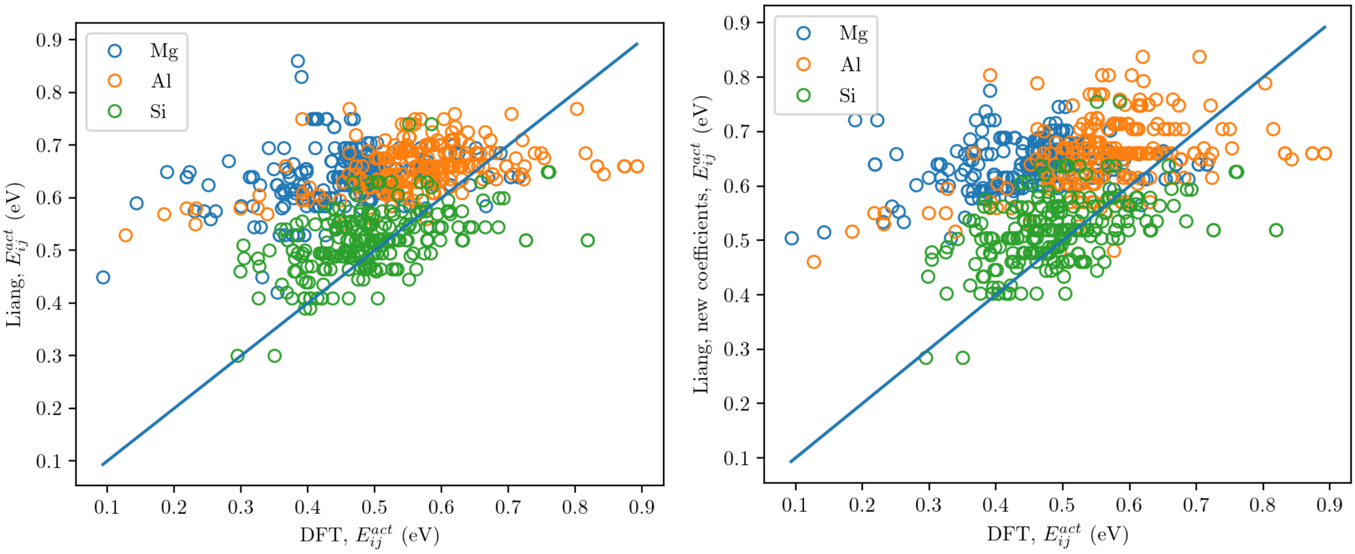
4.2.3 BEP-model

The BEP model briefly introduced in section 2.5 was tested on the dataset from the DFT calculations. The coefficients for equation (2.40) was fitted using the linear regression package from SciKit-learn[2] and the BEP relation was benchmarked against DFT calculations of the activation energies. The plot of the results for the BEP relation for Al, Mg and Si is shown in figure 4.5

[2] Scikit-learn is a free python library for Machine Learning and is found on https://scikit-learn.org/stable/

Interaction coefficient	Liang	DFT
ϵ_{Mg-Vac}	-0.015 eV	-0.0103 eV
ϵ_{Si-Vac}	-0.025 eV	-0.0446 eV
ϵ_{Mg-Si}	-0.04 eV	-0.02583 eV
ϵ_{Mg-Mg}	0.04 eV	0.01441 eV
ϵ_{Si-Si}	0.03 eV	0.01416 eV
$\epsilon_{Vac-Vac}$	0	-0.00012 eV

Table 4.5: Table showing the interaction coefficients from article by Liang et. al [4] compared to interaction coefficients calculated using equation (2.24) from DFT calculations.

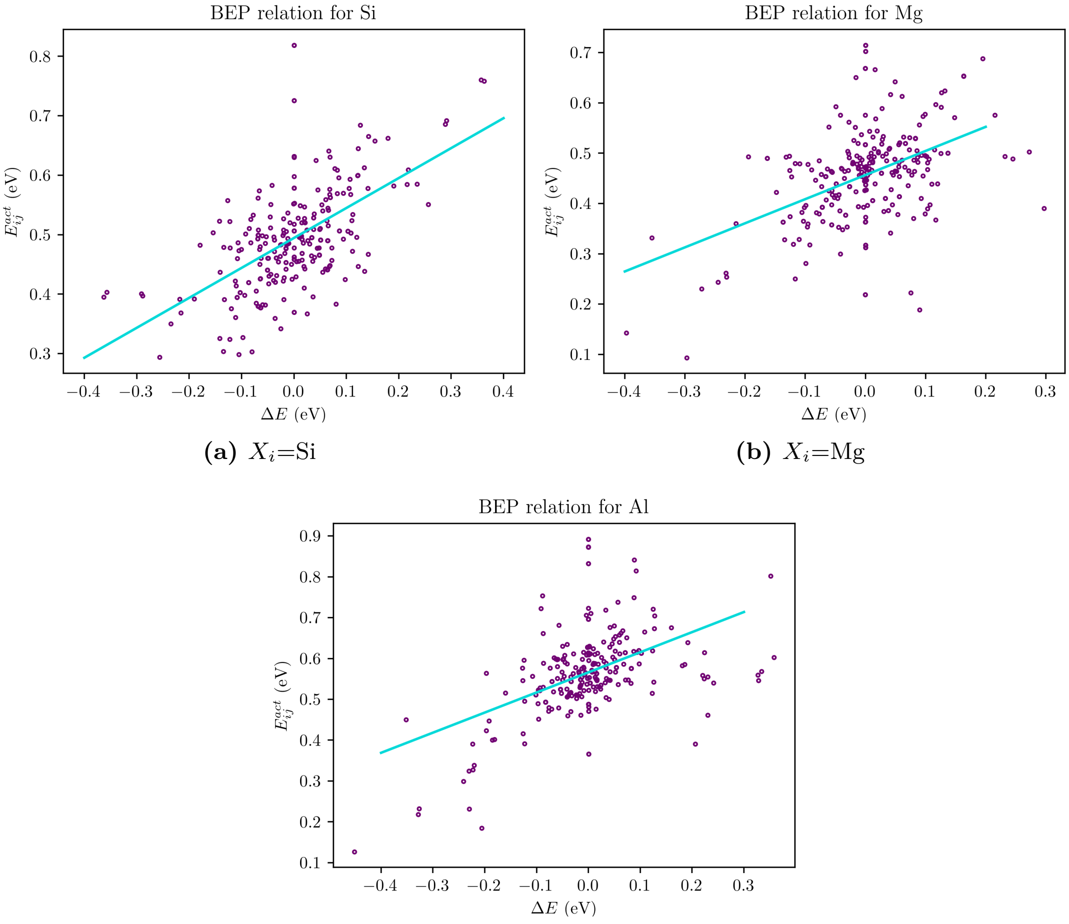


(a) **(b)**

Figure 4.4: Plot benchmarking E_{ij}^{act} calculated using equation (2.23) against E_{ij}^{act} calculated from DFT calculations. E_{ij}^{act} is calculated with interaction coefficients a) Liang and b) DFT in table 4.5.

and the RMSE values for each are presented in table 4.6 along with the fitted coefficients from equation (2.40). If the activation energy can be modelled using a linear relationship with ΔE, the activation energy is following the BEP relation.

The plot in figure 4.5 shows on average an increase of E_{ij}^{act} when increasing ΔE as the BEP model predicts, however the variance in the activation energies is large. Higher or lower activation energies with small energy differences are particularly inaccurate using this model.



(a) X_i=Si

(b) X_i=Mg

(c) X_i=Al

Figure 4.5: Plot of the DFT calculations of E_{ij}^{act} on y-axis and energy difference for the initial and final configuration for a jump ΔE on the x-axis for a) Si, b) Mg and c) Al. The blue line is the BEP model from equation (2.40) fitted to the data using coefficients presented in table 4.6.

X_i	E_0	k	RMSE
Al	566.05 meV	492.33 meV	8.10 meV
Mg	456.56 meV	479.22 meV	6.63 meV
Si	494.42 meV	503.68 meV	4.87 meV

Table 4.6: Table of the resulting RMSE for the BEP relation atom type X_i jumping to the vacant position. The coefficients E_0 and k from equation (2.40) are found using linear regression on the dataset.

The lack of linearity between ΔE and E_{ij}^{act} indicate that the BEP model does not do well in describing the barriers. The BEP-model assumes the local contributions to the activation to be constant for each jumping atom, thus the results for BEP used in this case shows that the activation energies are dependent on local contributions and not only differences in the total energy and a constant. This is also supported by literature for binary alloys where *ab inito* calculations has shown strong sensitivity to local changes in Al-Li and Al-Cu alloys [48]. The solutions to this is to include local variations by using methods like KRA.

4.2.4 KRA

The method of the activation energy for KRA from equation (2.39) is split into two parts in the SKMC as described in section 3.2 and is first presented as two separate results in this section. The first part of the results will be the the process of fitting Q, the average barrier from the forward and backward jump from equation (2.38). The second part will be the results of the CE of the total energy from equation (2.28) that is presented and compared to an open source CE software. The two parts will be combined to (2.39) and benchmarked to the DFT calculations for E_{ij}^{act}.

Fit of Q

The fitting process of Q is similar to the forward activation barrier, and the clusters defined are the same as used for the forward activation energy defined in table 4.2. The matrix containing the data for the configurations found in the dataset, $\mathbf{X}$, is therefore the same. The vector containing the $\vec{y}$ is defined as the average barrier, Q, of the jump.

The same prior as for the forward activation energy from equation (2.36), $Q_{\alpha\alpha}$, is used for this fitting process, and equal input variables for the fitting of hyperparameters as in section 4.2.1 was used with 10 000 attempts. The result of fitting the hyperparameters are shown in table 4.7.

Value	CV	RMSE	Hyperparameters
Q	20.11 meV	5.14 meV	$6.723 \cdot 10^{-20} D_\alpha^0 + 0.1172\gamma_\alpha^2 + 7.279 \cdot 10^{-19}$
ΔE CE	78.90 meV	58.78 meV	–
$E_{ij}^{act}(\Delta E$ from CE)	–	38.71 meV	–
$E_{ij}^{act}(\Delta E$ from IC)	–	33.19 meV	–

Table 4.7: Table presenting the result of the fit for the average activation barrier Q, ΔE and presents the CV score, RMSE and the hyperparameters used for fitting the ECIs. The two latter entries are the resulting E_{ij}^{act} from equation (2.39) using ΔE from CE and the interaction coefficients from 4.5

The hyperparameters in table were used to fit the ECIs and the resulting CV score using LOOCV and RMSE is presented in table 4.3. A notable difference from the hyperparameters for Q is the power order of D_α which is 0.

This means that the best fit for hyperparameters is independent on the distance to the jump atom which might stem from the average barrier should in reality be dependent on the transitional state, the state between i and j. This is further discussed in section 4.6. The RMSE and CV of Q is seen to be lower than of E_{ij}^{act}, which is not to expected if each jump was present only once. However, the datasets consist of both the forward and reversal jump which contributes to each Q to appear twice in $\vec{y}$. The result of the cluster expansion of Q is presented in figure 4.6, where the predicted values are plotted against the average of $(E_{ji}^{act} - E_{ij}^{act})/2$ from the DFT calculations.

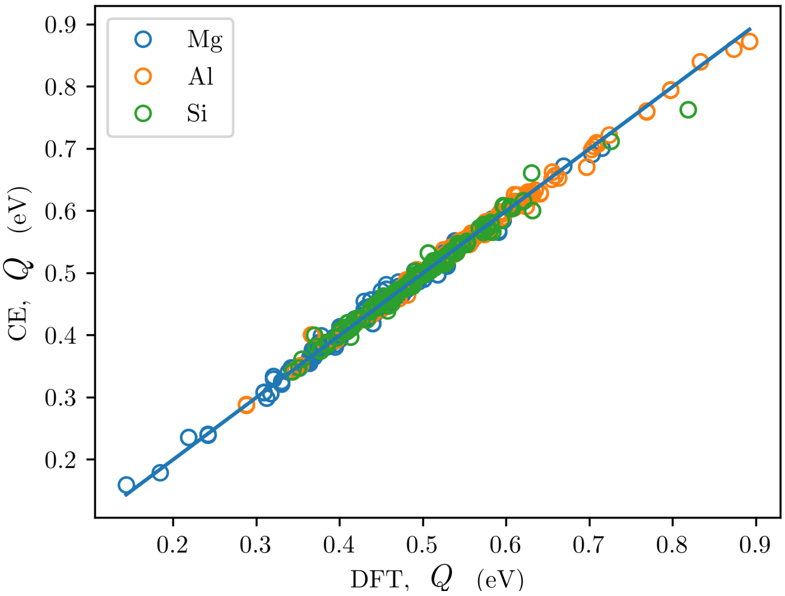


Figure 4.6: Benchmark of the predicted Q from equation (2.2) fitted using CE with the hyperparameters in table4.7 against Q calculated from DFT.

Fit of E^{tot}

The fit of E^{tot} is done by using the same dataset as used in the fit of Q and E_{ij}^{act}, but only the configurational energy for the system is used. For this the POSCAR files using the $3 \times 3 \times 3$ super-cells are used in order to remove the Al atoms that does not contribute to the total energy. The clusters that are used to find ECIs are single-sites and 1st, 2nd and 3rd nearest neighbour pair configurations. For this fit the dataset was split in to test data and training data, with the training data only used for The result of fitting total energies using the expression for cluster expansion equation (2.28) is presented in table 4.3 and the predicted values are benchmarked with both test data and training data in figure 4.7.

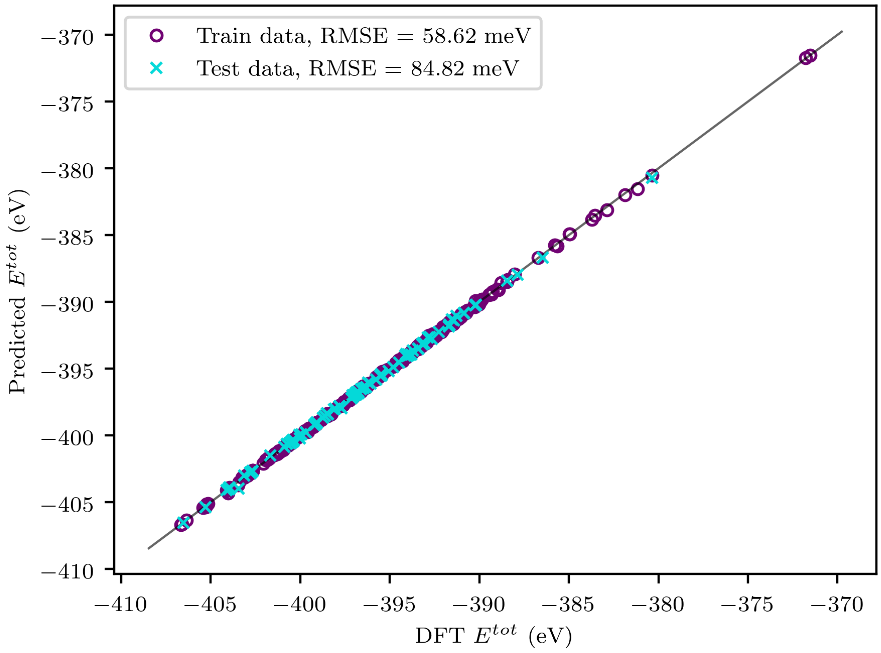
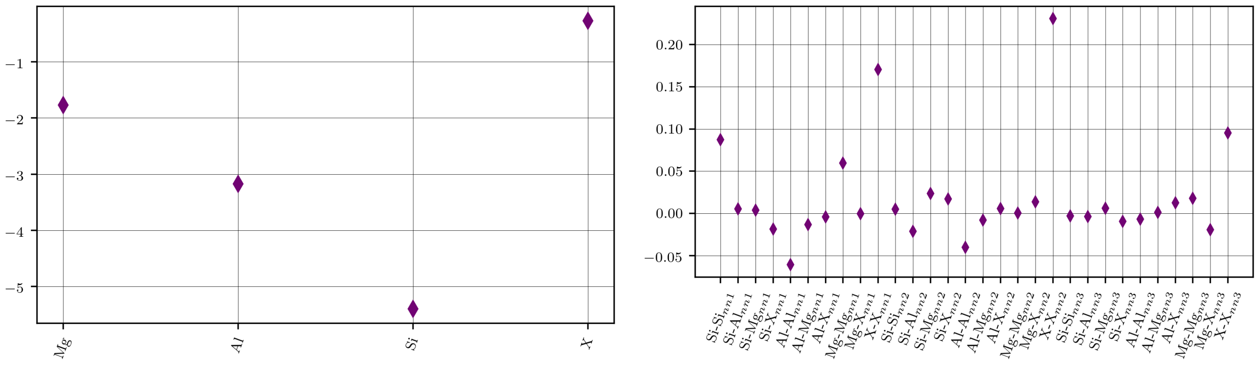


Figure 4.7: The predicted E^{tot} using the CE of the total energy from equation (2.28).

The coefficients from the CE^{tot} are plotted in figure 4.8. Figure 4.8 a shows the ECIs corresponding to the single-sites in the configuration. Figure 4.8 b shows the ECIs for the pair clusters defined on the x-axis. The single-site clusters are seen to be up to hundred times larger than the pair coefficients, indicating that most of the energies is explained by what atom type that is present in the system. The energy contribution the single-sites has on the system is in magnitude the range of 1- 5 eV per atom. The lowest coefficient is seen to be from Si, indicating that the configurations with increased amount of Si atoms have a lower total energy. The vacancy the highest value is the vacancy, but the real effect of a vacancy in the system is hard to tell due to there being maximum two vacancies in the system at all time. The pair coefficients is ranging between -0.05 meV and 0.23, where pair of vacancies is seen to have the largest coefficients.

The cluster expansion of the total energies, CE^{tot}, is benchmarked against CLEASE, a Python package created by David Kleiven, PhD Candidate at Department of Physics at NTNU [49]. The coefficients used for energy calculations using CLEASE are fitted using CE on DFT calculations performed with the DFT Python code GPAW[3]. The configurations used for the DFT

[3] More information about the GPAW is found at https://wiki.fysik.dtu.dk/gpaw/



(a) **(b)**

Figure 4.8: Plot of ECIs from the fit of (2.28). Figure a) shows the single-site coefficients and figure b) shows the pair coefficients from the 1st, 2nd and 3rd neighbour.

calculations are combinations of Mg, Si, Al and vacancies. The expansion includes clusters up to 4 sites and maximum cluster size of 5Å. The resulting model have a CV score of 6 meV/atom.

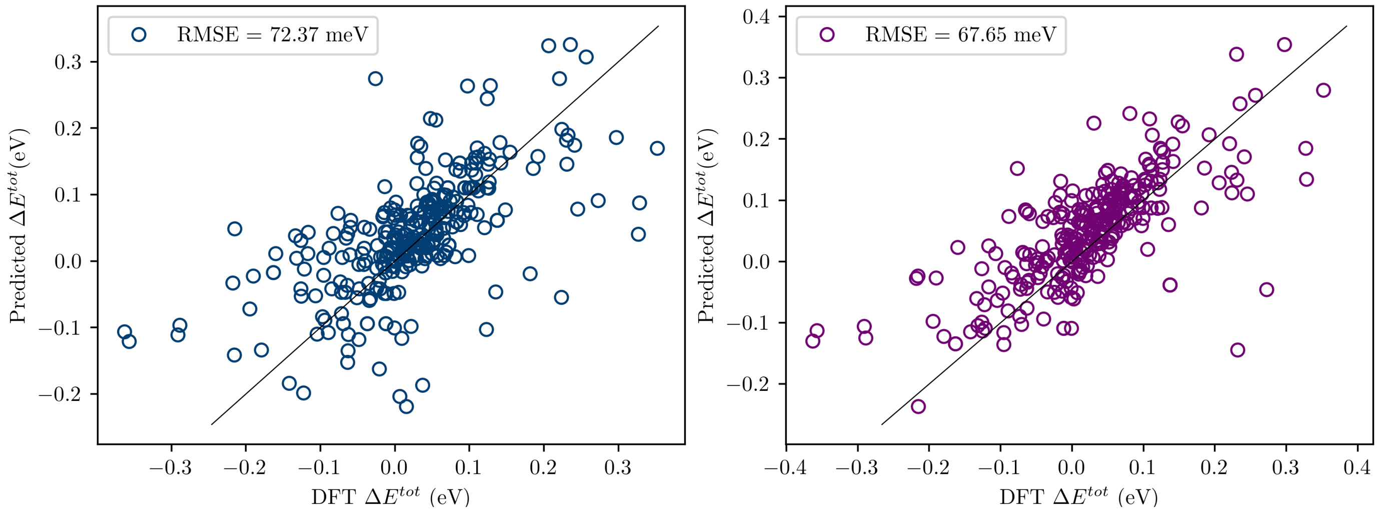
The total energy difference, ΔE, between the adjacent states before and after a jump is benchmarked against the DFT calculations in figure 4.9 a. The RMSE is of 72.37 meV, which is relatively large considering that ΔE is ranging between -380 meV and 390 meV. The corresponding benchmark with CLEASE is shown in figure 4.9 b and has a lower RMSE of 67.65 meV however still large relative to the order of ΔE. CLEASE is using more coefficients by including three and 4 sites compare to the maximum order of 2 using the in-house CE. This might be the source of the increased accuracy of this model. It's been proven hard to increase the accuracy further of the values of ΔE, which might be due to limitations from the DFT calculation of 108 atoms.

The prediction for the DFT calculations of E^{tot} using both the in-house and CLEASE version of CE is seen in figure 4.10. Figure 4.10 a) shows the values are predicted on the line, however the CLEASE prediction seen in 4.10 b is seen to have a slight offset of ~2 eV. Since the shift is consistent for the various structure, the error most likely stems from the coefficient of Al, since Al is the atom that has the largest occurrence in all the data entries.

The activation energy

Using the the Q and ΔE from the fitting procedures E_{ij}^{act} is calculated and the result is benchmarked with the DFT calculations in 4.11. The resulting RMSE is found in table 4.7. ΔE has a RMSE that is 10 times as large than the fit of Q which introduces the errors seen in E_{ij}^{act}.

A different approach of calculating the energy difference was to utilize the same principles as Liang [4] for calculating ΔE. The assumption is that

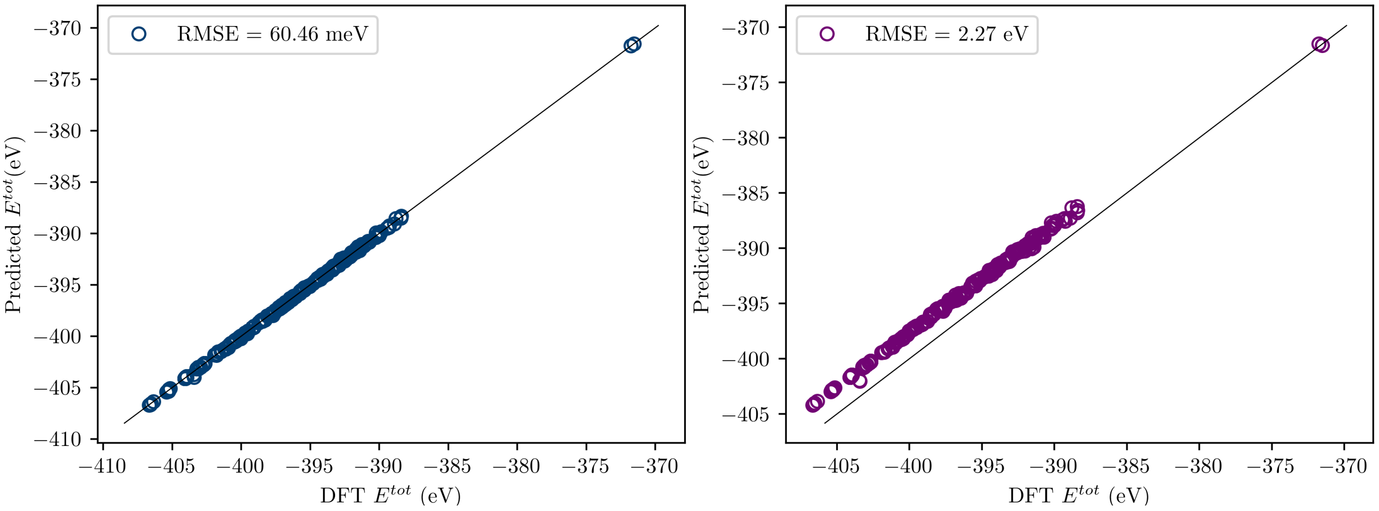


(a) .

(b)

Figure 4.9: The predicted ΔE between two adjacent states by using CE and benchmarked against the DFT calculations. Figure shows the energy differences for a) the in-house CE and b) the CE using CLEASE [49].

e



(a)

(b)

Figure 4.10: The predicted E^{tot} for the dataset by using CE and benchmarked against the DFT calculations. Figure shows the energies for a) the in-house CE and b) the CE using CLEASE [49].

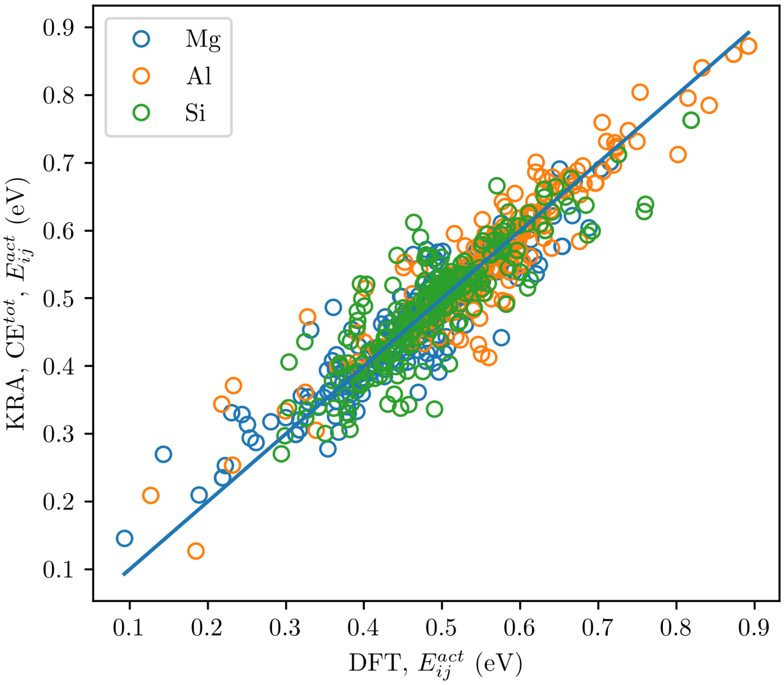


Figure 4.11: Benchmarking of the activation barrier using KRA from equation (2.39) with ΔE calculated from using CE of the E^{tot}.

the expression for E_{ij}^{act} that corresponds to the change in state is able to approximate ΔE between the two adjacent states. The ΔE is thus calculated from the expressions using the same variable as in the two parenthesis in equation 2.23:

$$\Delta E = -\left(\sum_{k\in NN_j} \epsilon_{X_k Vac} + \sum_{k\in NN_i} \epsilon_{X_k X_i}\right) + \left(\sum_{k\in NN_i} \epsilon_{X_k Vac} + \sum_{k\in NN_j} \epsilon_{X_k X_i}\right) \tag{4.1}$$

The coefficients used for the method are the DFT calculated interaction coefficients given in table 4.5. The activation energies using this method is plotted in 4.12 and the RMSE can be seen in table 4.7. The RMSE for this method 5.5 meV lower compared to the KRA method using CE^{tot}, which is high compared to the errors from the fit of Q.

4.3 Simulations

This section will go through the simulation results for using KRA and CE method using the ECIs found by the fitting in section 4.2. They will compared

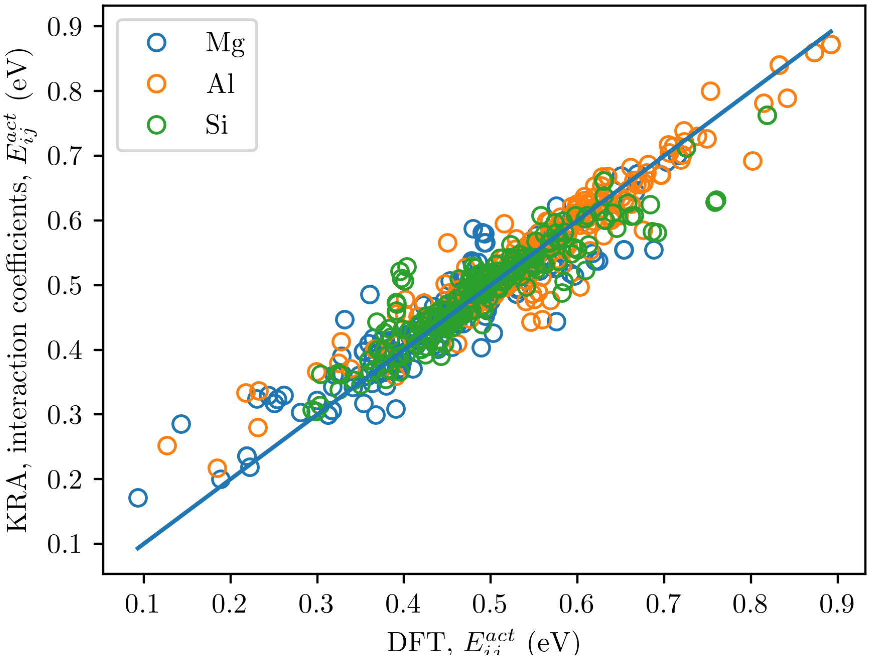


Figure 4.12: Benchmarking of the activation barrier using KRA from equation (2.39) with ΔE calculated from interaction coefficients in table 4.5 against E_{ij}^{act} calculated from DFT.

against the Liang [4] method which was the originally implemented method for simulating the Al-Mg-Si alloy in SKMC. The comparisons are done by examining the clustering of the system, the energy evolution and the diffusitives.

The parameters used to initialize the simulation is summed up in table **??**. The equilibrium values for the formation enthalpy, H_V^F, and formation entropy, S_V^F is taken from the article of Carling et al. [23]. The values for attempt frequency ν, is set as the same as Liang et al. article [4] presented in table 2.1. Simulations are run in a 25 × 25 × 25 FCC lattice with lattice parameter $a = 4.05$. The added solute percentage is 0.67 % Si and 0.77 % Mg, which is similar percentage to the Al6060 alloy. The system will be simulated in the natural ageing phase of the ageing process, and is therefore set to be $T = 293.15$K. All simulations are run using one core and has been run on the cluster project IDUN [50] and the Linux cluster for Department of Physics, NTNU.

The simulation will mostly be focused on four different runs and all methods are run for 10^7s. The method used in the article by Liang et al. [4] is run using the coefficients in table 2.1 and will be referred to as the Liang run. The cluster expansion method will be run using the coefficients from the fit in table 4.3 and will be referred to as the CE run. The kinetically resolved

Parameter	Value (unit)
k_B	8.6173 (eV/K)
T	293.15 (K)
a	4.05 (Å)
Si (%)	0.77
Mg (%)	0.67
H_V^F	0.66 (eV /K)
S_V^F	1.6 k_B (eV/K)
C_V [a]	$2.23 \cdot 10^{-11}$
ν_{Si}	1.57 $\cdot 10^{13}$ (s^{-1})
ν_{Mg}	1.86 $\cdot 10^{13}$ (s^{-1})
ν_{Al}	1.66 $\cdot 10^{13}$ (s^{-1})
Size	25 × 25 × 25 (unitcells)

Table 4.8: Initial parameters for simulations presented in this section.

[a] For Liang method and Liang method with new coefficients this is set at $1.41 \cdot 10^{-4}$ due to continuity to previous work.

activation barrier method is run using the cluster expansion for the total energy and using interaction coefficients from 4.5 to calculate the total energy difference. The benchmark for both methods are given in 4.7 and are referred to as KRA_CE and KRA_IC run respectively.

4.3.1 Clustering

This subsection will present the time evolution of the clustering observed during simulation of the activation energy methods. The clustering of the system is measured by the number of solutes in each cluster, and the clusters are defined as configurations of solutes connected through nearest neighbours. For the results presented in this section, the threshold for being a cluster is size 2 or larger. The small threshold is chosen to be able to compare evolutions with little to no clustering.

Liang method with new interaction coefficients

The calculation of the new interaction coefficients was used tested in a simulation using the same parameters as the Liang method, but using the interaction coefficients in table 4.5. The resulting evolution of cluster size during the simulation is plotted in figure 4.13 and compared with a run using the interaction coefficients used by Liang. The evolution is observed to increase for both meth-

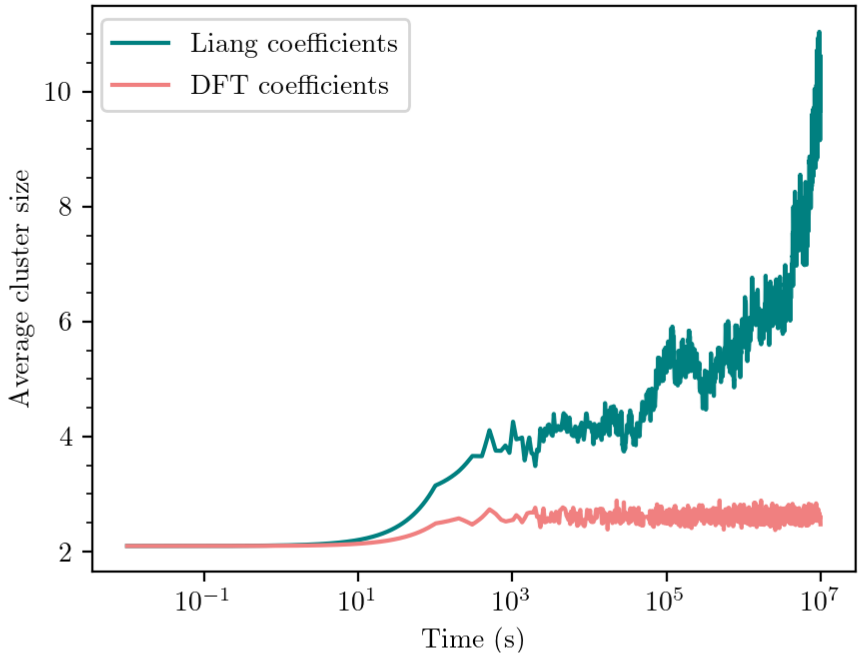


Figure 4.13: Time evolution of the clusters, defined as 2 or more solute as nearest neighbours, of the Liang method [4] using interaction coefficients from Liang and DFT presented in table 4.5 and the rest of the parameters from 2.1.

ods until about 100 seconds into the simulation, where the average size of the run using interaction coefficients from DFT is seen to stagnate and fluctuate between 2 and 2.5 sites per cluster. The run with Liang coefficients is seen to increase in average cluster size throughout the run.

In the article by Mørtsell et al. [9] that uses bulk calculations of 500 atoms, it was found interaction coefficients between the solutes to be even lower in magnitude than the coefficients found using 108 atoms, which indicates that smaller cells could overestimate the impact of certain interactions. This might be the cause of the clustering seen using Liang method. Having Mg-Mg and Si-Si having a large repulsion and Mg-Si having a large attraction would make structures of altering layers of Mg and Si overestimated favourable. Snapshots of clusters formed are further presented in next subsection.

Clustering of the methods

Figure 4.14 shows the evolution of the average cluster size and the number of clusters per volume found in the four different simulations. The Liang method is already known to form clusters [4][3] and is used as an example of clustering to be compared to the rest of the methods. The plot in 4.14 a has addition to the average number of solutes in the cluster the number of defined cluster per volume. After 10^7s, the average cluster size has reached ∼11 solutes and the cluster number density is decreasing. The decrease of number of clusters in the system is happening because the clusters are accumulating to the larger cluster and the system is limited to 900 solutes. A snapshot of the the positions of the defined clusters at 10^7s can be seen in figure 4.15 a. The large cluster of alternating Mg and Si seen in the figure is the stable $L1_0$, structure, further studied in the works of Kleiven and Akola [51].

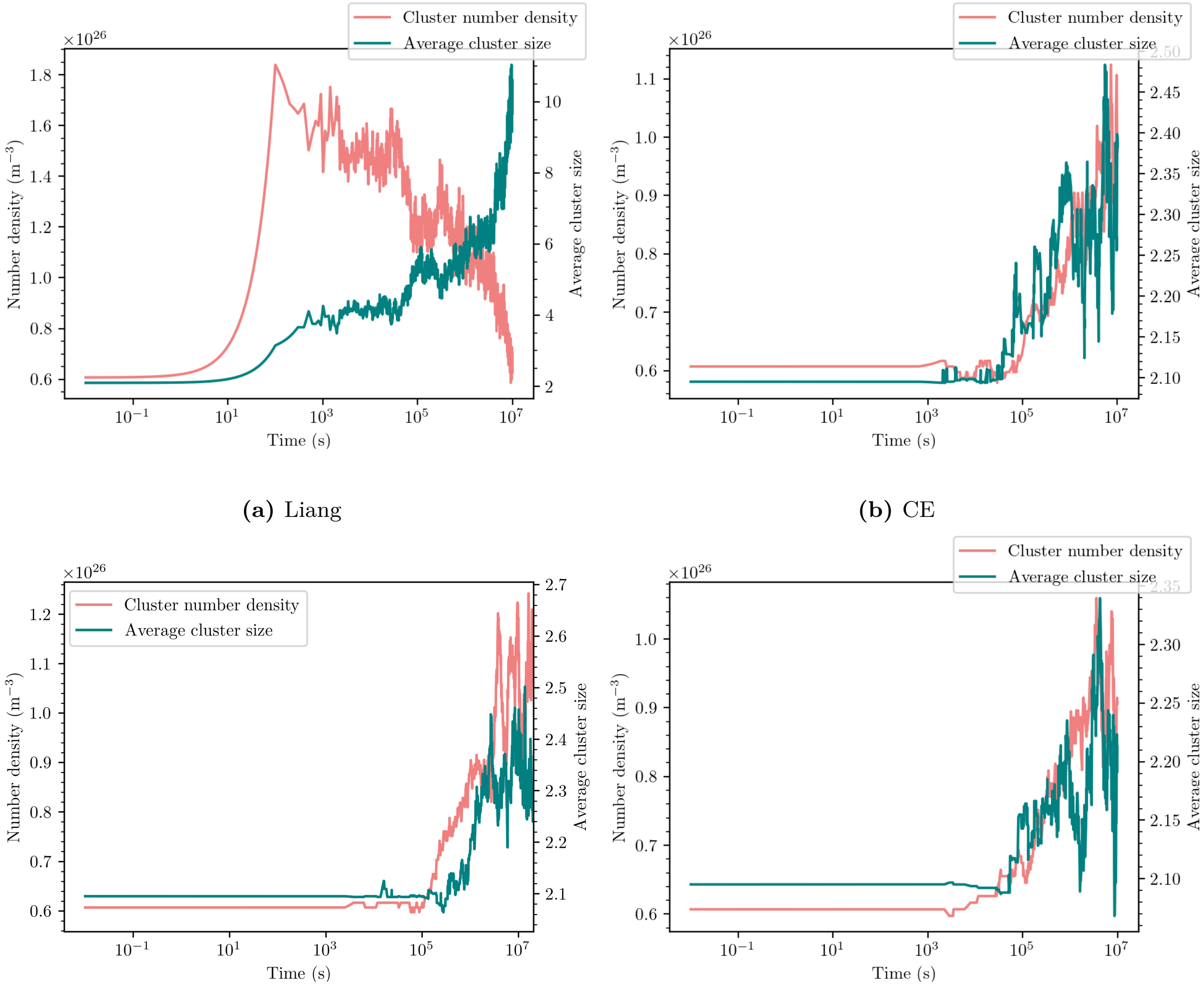
The average cluster size and the cluster number density for the KRA and CE methods are seen in figure 4.14 b, c and d. All three methods are seen to have similar cluster evolutions, where the average cluster size is seen to increase to 2.3-2.7 solutes per average where it fluctuates for the rest of the simulation. The cluster number density is seen to have similar evolution, increasing until it reaches an equilibrium and continue the fluctuating behaviour. Snapshots showing the position of the clusters for CE, KRA_CE and KRA_IC at 10^7s are seen in 4.15 b, c and d, and shows that the are dispersed randomly in the system and rarely above the size two which is the threshold for being observed in this figure. This behaviour shows that these three methods does not have the clustering effects, essentially having activation energies that do not see clusters of Mg and Si as a favourable structure in the system.

4.3.2 Diffusivities

This section will present the diffusitives logged during the simulations using the different methods of calculating activation energy. The diffusitives logged during the simulation is the average diffusivity for all the atoms of atom type X_i. The diffusivity is logged as [19] $D_X = \lim_{t\to\infty} \frac{\langle |r_{\vec{X}}(t) - r_{\vec{X}}(0)|^2 \rangle}{6t}$, with $r_{\vec{X}}(0)$ being the initial position of the atom, $r_{\vec{X}}(t)$ the position after simulated time t.

The results for the diffusivity for each run is plotted in figure 4.16. As observed in previously works [3], the diffusivities for the Liang run seen in figure 4.16 a, is of the order $10^{-25} - 10^{-25}$ after 10^7s and is decreasing approximately 3 orders of magnitude during the simulation. The decrease in diffusivity is most likely due to the solutes getting trapped in clusters, and the amount of available and moving solutes decreases as the number of solutes in clusters increase. This behaviour happens because the simulation system has a limited amount of atoms available. This might impose a problem if the simulation was to run longer, as the solutes would be inaccessible for the system. Since the current goal is simulating the early precipitation process, this will not affect the simulations used for this purpose.

The diffusitives for the CE and KRA methods in figure 4.16 b, c and d is observed to not decrease, and are 4 times order of magnitude smaller than the Liang run. The magnitude is lower due to C_V being chosen to be the same as in



(a) Liang

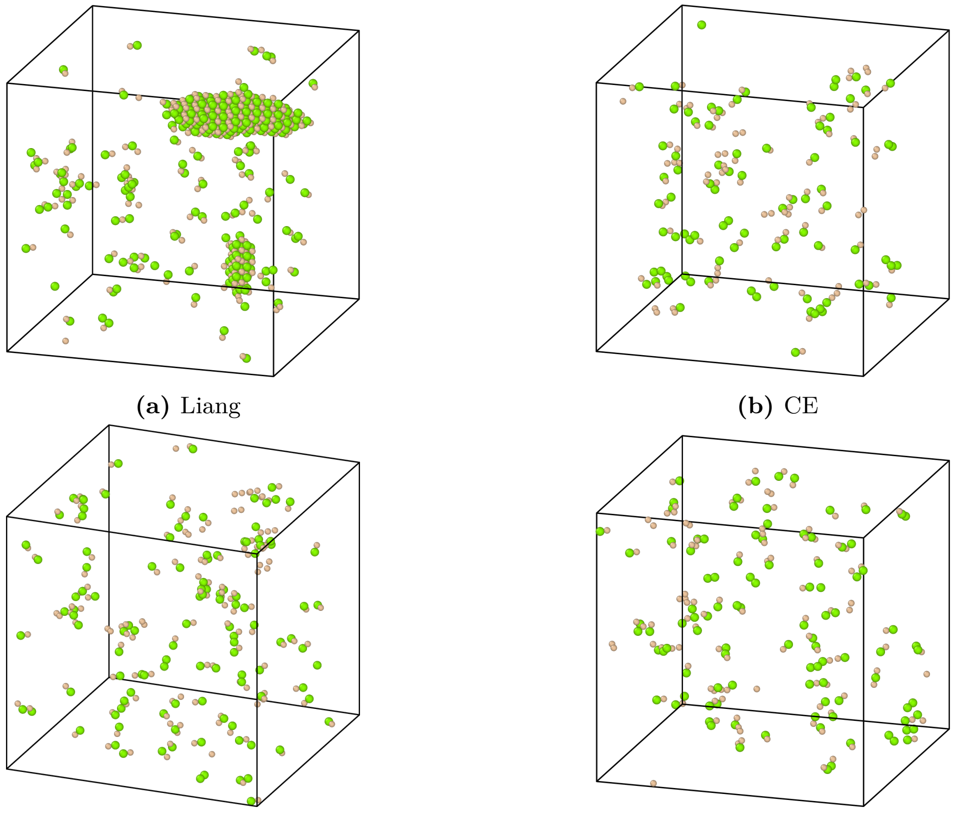
(b) CE

(c) KRA_CE

(d) KRA_IC

Figure 4.14: Evolution of the average cluster size for clusters of size two or larger. Logged as the average number of solutes in each cluster and the number of clusters per volume.

Liang article, and $t \propto 1/C_V$ is scaling the diffusivity. The diffusitives for Liang and CE are further discussed in the works of Nygård [3]. For both the forward CE seen in figure 4.16 b and the Liang 4.16 a, the diffusivity for Si is largest, followed by Mg and Al. This differs from the two KRA methods where D_{Si} and D_{Mg} is observed to be lower and of similar order. For KRA_CE, D_{Si} is seen to be become lower than D_{Mg}, which differs from the rest of the runs. The values from Mantina et al. [52] seen in table 4.9, however, is seen to have D_{Si} be larger than D_{Mg} indicating that the lowering of this value might be wrong.



(a) Liang

(b) CE

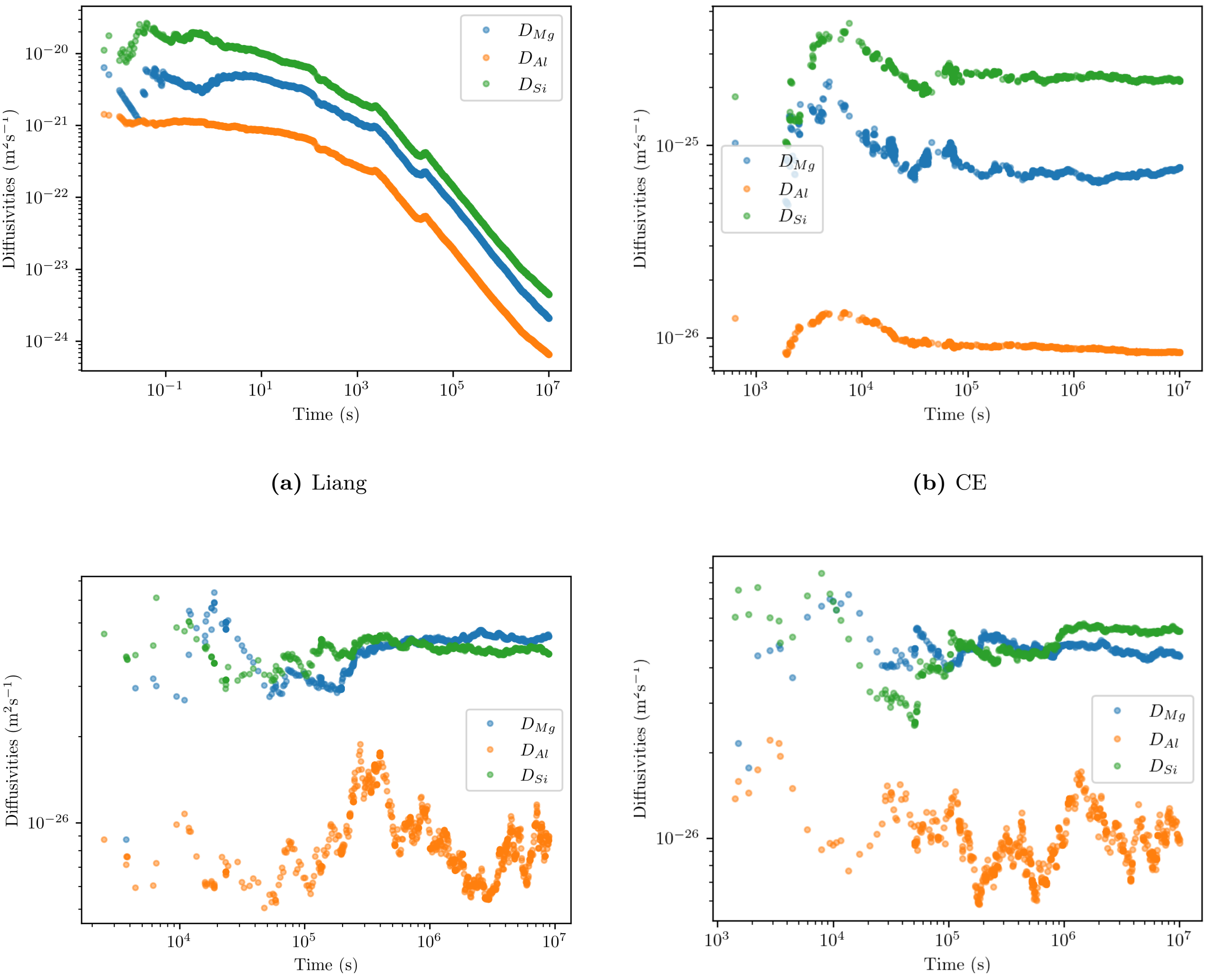
(c) KRA_CE

(d) KRA_IC

Figure 4.15: Snapshots of the clusters defined as 2 or more solutes connected as nearest neighbours at 10^7s.

The general lowering of D_{Si} using KRA methods indicates that the events with Si is chosen less frequently using KRA than Liang and CE. This could be happening because structures that are frequently seen in the simulation has higher barriers using KRA than CE. For example for the case of the activation energies for both KRA methods seen in figure 4.11 and figure 4.12, the E_{ij}^{act} for a Si jump with barrier $\sim 0.38 - 0.4$ is seen to have a larger predicted barrier than for CE seen in figure 4.2. Similarly, E_{ij}^{act} for Si jump have more of these overestimated barriers using KRA_CE, which could explain D_{Si} being even lower for this. None of the methods have the ratio between D_{Si} and D_{Al} as seen in [53].

For the KRA methods the diffusitivity for Al seem to be fluctuating during the run compared to the other two methods. The value of D_{Al} is also seen to be lower than CE and Liang.



(a) Liang

(b) CE

(c) KRA, ΔE from CE

(d) KRA, ΔE from interaction coefficients.

Figure 4.16: Diffusitives

4.3.3 Energy evolution

This section presents the energy evolutions of the KRA, Liang and CE method which done by presenting the logged values of the energy differences throughout the simulation. The logged energy will be plotted with a cluster expansions of the lattice at the time of the logged energy difference. The cluster expansion of the lattice is done using CLEASE [49] with coefficients presented in section 4.2.4. The energy logging is done using the difference in forward and

Value	Liang	CE	KRA_CE	KRA_IC	Løvvik et al. [53]	Mantina et al. [52]
D_{Al}	$6.58 \cdot 10^{-25}$	$8.42 \cdot 10^{-27}$	$1.20 \cdot 10^{-26}$	$9.64 \cdot 10^{-27}$	$8.67 \cdot 10^{-27}$	–
D_{Si}	$4.47 \cdot 10^{-24}$	$2.19 \cdot 10^{-25}$	$3.97 \cdot 10^{-26}$	$5.40 \cdot 10^{-26}$	$2.17 \cdot 10^{-28}$	$6.21 \cdot 10^{-26}$
D_{Mg}	$2.09 \cdot 10^{-24}$	$7.72 \cdot 10^{-26}$	$4.64 \cdot 10^{-26}$	$4.41 \cdot 10^{-26}$	–	$1.75 \cdot 10^{-27}$

Table 4.9: Table with the diffusivities in $\mathrm{m}^2\mathrm{s}^{-1}$ at the end of the KMC runs for Al, Mg and Si in room temperature T=293.15K compared to values in articles from Løvvik et al.[53] and Mantina et al. [52] .

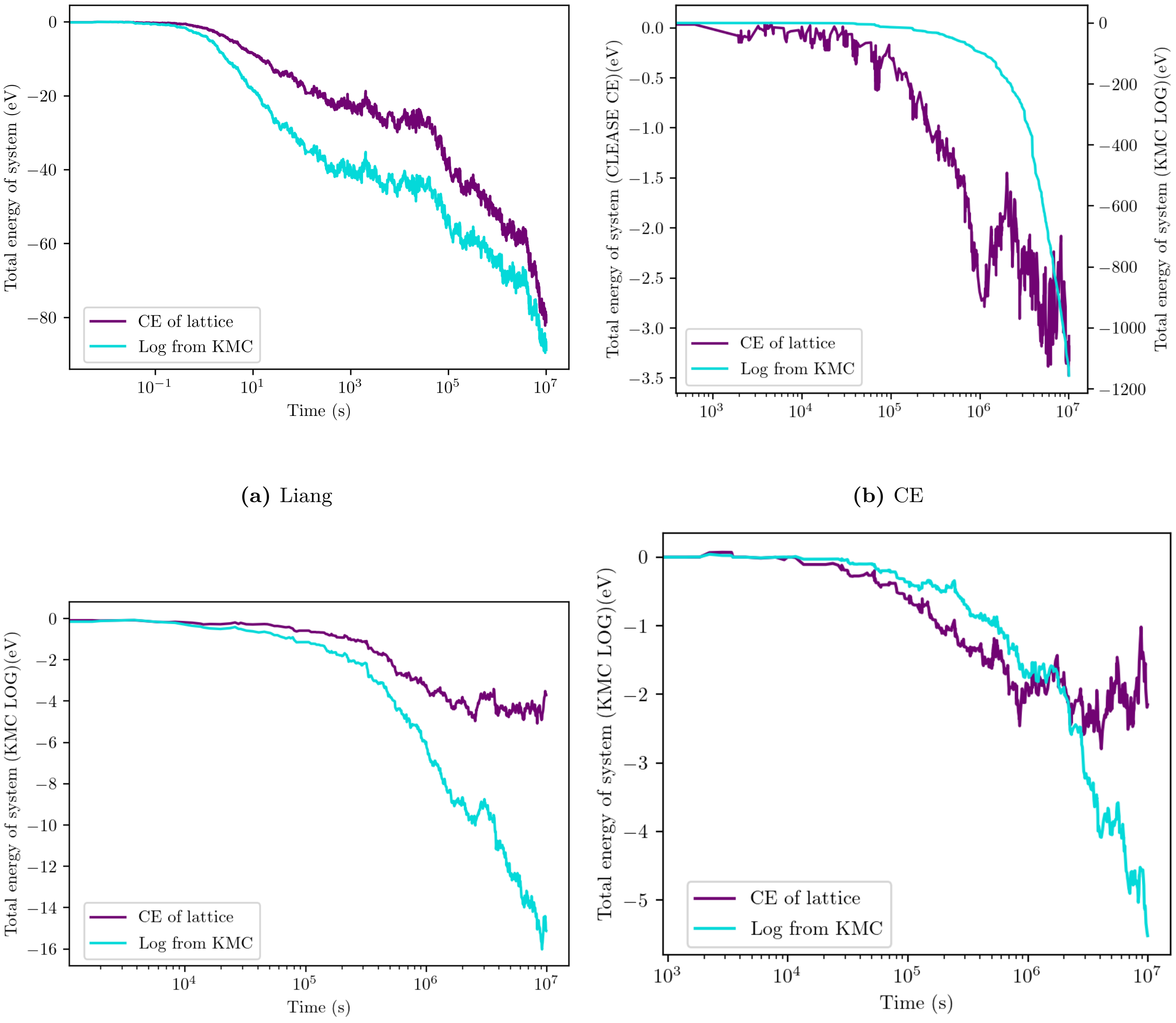
backward energy, $\Delta E = E_{ji}^{act} - E_{ij}^{act}$, for each iteration during the simulation. The start energy is defined as 0, so the energy logged are the difference from the start energy. The these energies are logged with the same parameters as the POSCAR files used to calculate total energy using CLEASE in order to calculate the differences of energies of the same time interval. To benchmark the methods, they are plotted against the total energy calculation of the system, calculated by CLEASE [49]. The cluster expansion of the total energy will be referred to as $\mathrm{CE}^{\mathrm{tot}}$ in order to avoid confusion with the CE method for the activation energy.

Figure 4.17 a shows the energy evolution using the Liang method. The evolution is similar to the calculated from $\mathrm{CE}^{\mathrm{tot}}$, but slightly shifted. The plot shows the energy from the log is reduced to -89 eV during the simulated time, ending up at a slightly lower energy than the energy calculated with $\mathrm{CE}^{\mathrm{tot}}$. The decrease in energy seen is due to more atoms being in clusters which are energetically favourable, which is seen to be reflected well in the activation energy of Liang.

In general for the plots 4.17 b, c, and d, the curve for total energy calculated with $\mathrm{CE}^{\mathrm{tot}}$ flattens, seemingly going to an equilibrium. This point corresponds to the point where the system stops clustering in figure 4.14 b, c, d at approximately $t = 5 \cdot 10^6$s, and is what is expected for a system that no longer changes in terms of number of clusters.

The evolution of a simulation using the CE run, seen in figure 4.17 b, shows the energy logged from the simulation to increase significantly from simulation start which was seen in the works of previous master student Nygård [3] using the smaller training set. The y-axis are separated due to differences in magnitude. The decrease from initial state is ~1900 eV for the logging and ~3.0 eV for the $\mathrm{CE}^{\mathrm{tot}}$ energy and corresponds to a difference of order ~600. If run for longer the drifting will increase exponentially while the $\mathrm{CE}^{\mathrm{tot}}$ will remain in the same order [3].

Using KRA methods seen in figure 4.17 c and d, the negative drifting energy is significantly lower than for the CE run. Both methods seem to drift further after the $\mathrm{CE}^{\mathrm{tot}}$ has come to an equilibrium. The logged energy from KRA_CE run can be seen to behave similarly locally to the calculated $\mathrm{CE}^{\mathrm{tot}}$, but systematically scaled by a drifting. For KRA_IC, the energy evolution is seen to be slightly underestimated during the beginning of the simulation



(a) Liang

(b) CE

(c) KRA_CE

(d) KRA_IC

Figure 4.17: Energy evolution of the different runs compared using the logged energy difference from KMC and cluster expansion of the total energy using CLEASE [49]. Note that the energy evolution for CE has two different y-axis due to very large differences.

before seen to continue to decrease after $\mathrm{CE}^{\mathrm{tot}}$ has stopped decreasing.

The decrease of energy drift seen by introducing the KRA method implies that the source of this drift are errors from not being able to represent ΔE in the cluster expansion. Introducing the ΔE explicitly corrects for this, and even though the expression for ΔE introduces some errors to the fitted activation energies, the KRA method is seen to be better at representing the systems energy evolution.

Seeing both KRA methods have some drifting is most likely because the CE of the average activation energies is not fully independent of the jump direction. The clusters defined in the implemented CE^{act} is relative to the jump atom even though the value it is fitted to is the average. This makes the average still be dependent on if the jump is from state j or from state i in figure 2.12. Preferably should the clusters be defined from the transitional state at E_{saddle} as the distance from the middle of the jump. This is further discussed in section 4.6.

4.4 Forming precipitates

The implementation of being able to jump to the octahedral sites was done by demanding that certain criteria were met for a jump to happen, and the criterion for being an active octahedral site was set to be an octahedral site surrounded by 4 Si atoms in the same plane, one vacancy as a nearest neighbour and Mg as a jump atom. Using this requirement on a simulation using CE resulted in 13 jumps into octahedral sites out of 10^9 jumps total. These jumps were seen to be unstable and jumped out shortly after. The configuration where the octahedral jump was seen to happen the most is seen in figure 4.18 a. Figure 4.18 b shows the configuration where the Mg has jumped into the octahedral site.

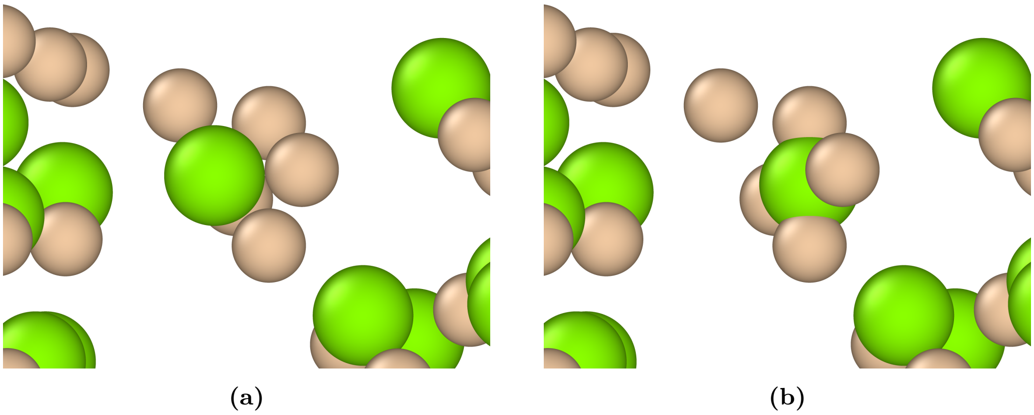


Figure 4.18: Snapshot showing a jump into an octahedral site between a layer of Si. The Al are not included in the snapshot to increase visibility but is present in the simulation.

This octahedral jump happens regardless of the activation energy for the jump into octahedral, and it is therefore not physical. The current configuration of only a layer of Si and 1 vacant position where the jump is seen to happen most frequently is not seen to happen using DFT calculations. The configurations that lead to octahedral jumps to happen in the DFT calculations which is described in section 2.1.2 is not seen to occur using simulations using CE or KRA.

The initial belief that the jump needed an extra vacancy in order to form was debunked with DFT calculations done by Inga G. Rindalen of 5 × 5 × 5 unit cells, where the β'' was able to be formed from no vacancy.

4.5 Residence time

The results from implementing the second order RTA is presented in this section and compared to the first order RTA originally used by SKMC. The average times spent doing one iteration for each method for calculating activation energy and using different orders of RTA is presented in table 4.10. The time in this table is calculated as the time spent by the CPU on performing one KMC step. The CPU used for each simulation is on a Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, and each is run using one core. The simulations are run using no observers to remove variables that might affect the time, and the initialization of the simulation is not included. As expected, and observed earlier [3], the Liang method is faster than both CE and KRA, and using the second order RTA increased the time spent for each iteration ∼10 times for Liang and ∼12 times for CE and KRA. Time spent on each iteration increased approximately 620% from Liang to CE, and increased ∼7% from CE to KRA. The increase of time from Liang to CE is due to the increase of calculations of coefficients from nearest neighbours, to searching and calculating configurations for a neighbour list including up to fifth nearest neighbour. KRA method increases the time by needing to calculate the total energy difference, but by using only the changes related to the neighbours of the jump atom and vacancy, only the coefficients related to first, second and third neighbouring sites of the vacancy and jump atom is included. This affects the simulation less than the implementation of CE.

The time taken to calculate the transition rates is expected to increase 12 times with the second order RTA since the second order also calculates the rates of the 12 neighbouring atoms for each possible jump, totalling in 12 × 12 = 144 rates. The fact that the time for each iteration increases less than 12 times for Liang is because each step in the KMC simulation includes more calculations such as choosing the event, updating coefficients and calculating the time-step, and the calculation activation energy using only nearest neighbours is corresponding to approximately 80% of the computing time for the first order. Since calculating the activation energies for KRA and CE is taking a larger fraction of the time for each step, the increased computing time for the second order will seen to be closer to 12.

Both Liang and CE was tested and compared using first and second order and the results are presented in the next subsections. Note that CE and Liang

Barrier method	RTA order	CPUtime/iteration (μs)	# iterations/second (s^{-1})
Liang	1	3	333 000
Liang	2	29	35 000
CE	1	187	5350
CE	2	2297	435
KRA	1	201	4975
KRA	2	2340	427

Table 4.10: Table comparing times spent on average for one step in SKMC using different methods for calculating activation barrier and orders of residence time. The simulations were done on one CPU core of Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz with no observers. The time for setting up the simulation is not included in the average time.

are run using different C_V and is not to be compared with each other in regards of simulation time.

4.5.1 RTA on the Liang method

In order to compare efficiency the first and second order RTA, two runs from each order of Liang is shown in figure 4.19. The simulation used for these plot are run with the previous parameters used for Liang and the end time of 10^5 s. Figure 4.19 a shows the fraction of every jump that is jumping back to previous site instead of the 11 other possible sites. From first order to second order, the number of reversals is seen to decrease from happening 75% of all jumps, to happening 25 % of all the jumps. Figure 4.19 b shows the percentage of total simulated time for the first and second order. The second order is seen to complete the simulation three times faster, despite the increase of computation time spent per iteration.

The smooth curve for both RTA orders in 4.19 b shows no sign of vacancy trapping that stops the system from evolving. This was not a problem reported for the Liang method and the reason is most likely due to smaller range in activation energies as seen in figure 4.4, which is preventing the system from ending up in an energy well.

To check if the diffusion process is altered by the new method, the diffusitives logged by KMC are compared and showed in figure 4.20. Both diffusitives is seen to decrease similarly, indicating both methods having the same diffusion dynamics regardless of the second order finishing three times as fast.

4.5.2 RTA on the CE method

The effect of the RTA method for the CE method is shown in figure 4.21. As seen in figure 4.21 a, CE using first order RTA has a larger fraction of reversals than Liang, up to 99% of all jumps. This means that 99% of each iteration

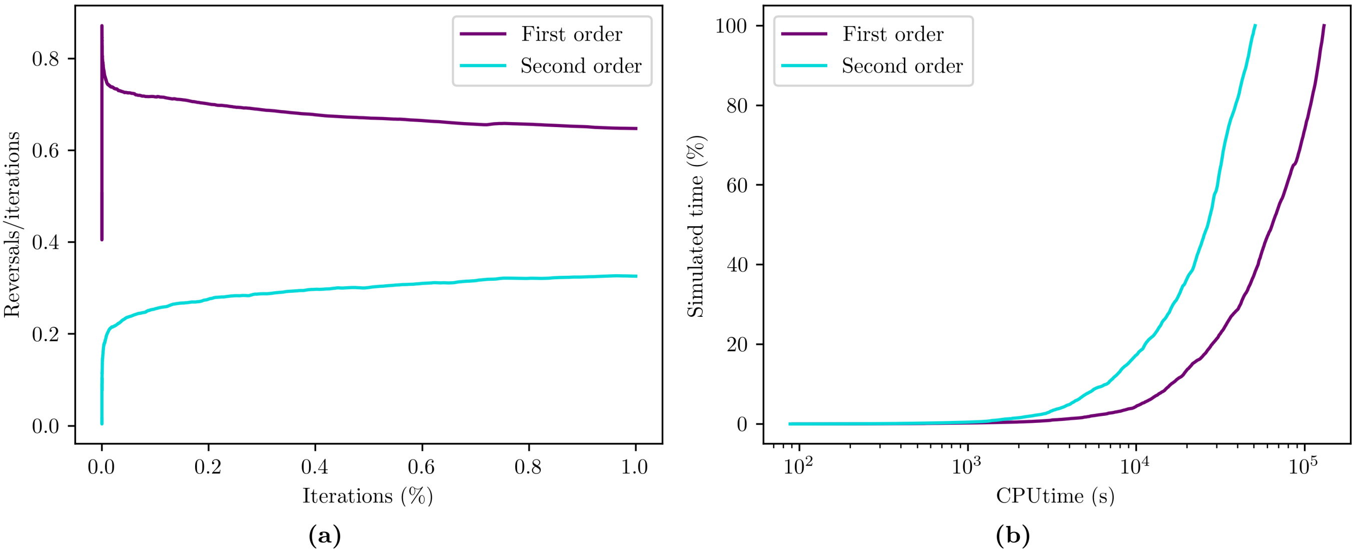


Figure 4.19: Simulation using the Liang [4] method and first and second order RTA. Figure a) plots the number of reversal per iteration, where 1 indicates all jumps are reversal. The x-axis corresponds to the percentage of total iterations done in the simulation. Figure b) shows the time spent by the CPU to complete the simulation.

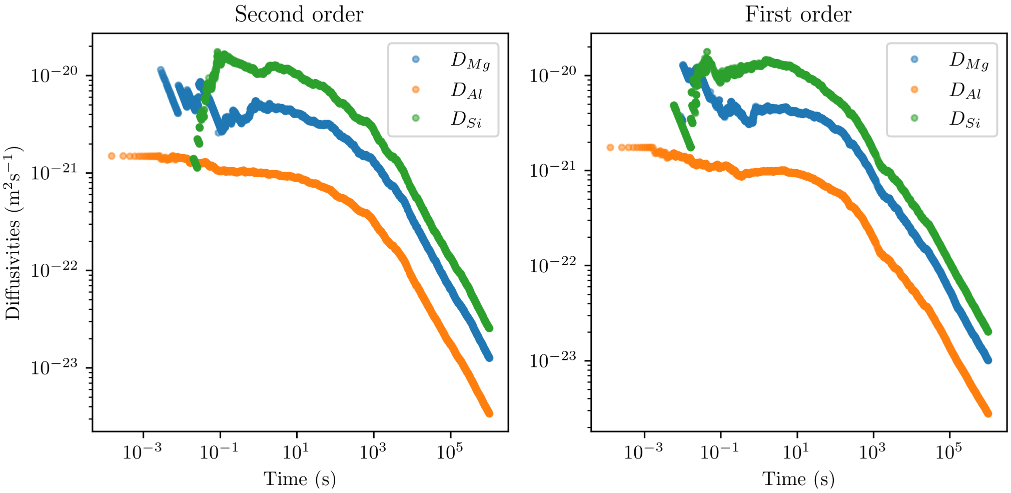
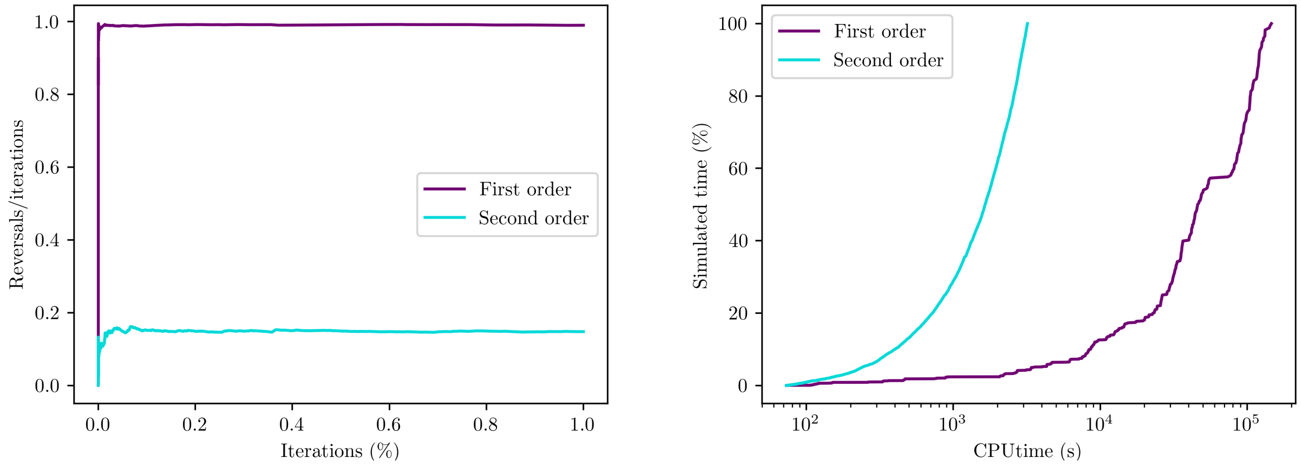


Figure 4.20: Plots of the diffusitives for Si, Mg and Al logged by KMC using first and second order RTA for the Liang method. [4]

is non-effective and does not allow the system to evolve. The second order decreases the reversal jumps to only occur 15% of all jumps. The resulting fraction of reversals is lower than seen for Liang using the second order method.

The explanation for this is that the probability to jump back, seen in equation 2.17, is scaled by the probability to escape when doing the reversal, β_j, and due to the probability being larger for escape using Liang method, the reverse jump is more frequently chosen.



(a) **(b)**

Figure 4.21: Simulation using the CE method and first and second order RTA. Figure a) plots the number of reversal per iteration, where 1 indicates all jumps are reversal. The x-axis corresponds to the percentage of total iterations done in the simulation. Figure b) shows the time spent by the CPU to complete the simulation. The plateaus seen from plot b) is the vacancy getting trapped and the system is stuck.

Figure 4.21 b shows the time evolution of simulation time for CE, where the second order shows an uninterrupted increase of simulation time while the first order shows plateaus where the system does not increase in simulated time. These plateaus are where the system is stuck and the vacancies are trapped jumping back and forward between few adjacent states. These trappings can be temporarily affecting the system, or in some cases, seen to be trapped for days until the simulation is stopped. The simulated time does not increase much in these traps since the rates are high and $t \propto 1/R_{N_e,i}$. The plot shows that these ineffective trappings of vacancy are removed when using the second order RTA. The time to complete the simulation was 50 times faster for the second order, not only due to eliminating reversals like Liang, but also due to the vacancy not getting trapped which was a frequently occurring problem with the previous implementation.

The comparison of diffusitives for the two RTA methods is seen in figure 4.22 and similarly to the Liang method, the second order is not seen to alter the diffusion process by the new rates regardless of being calculated 50 times as fast. This indicates that the second order RTA can safely be used without affecting the dynamics of the system.

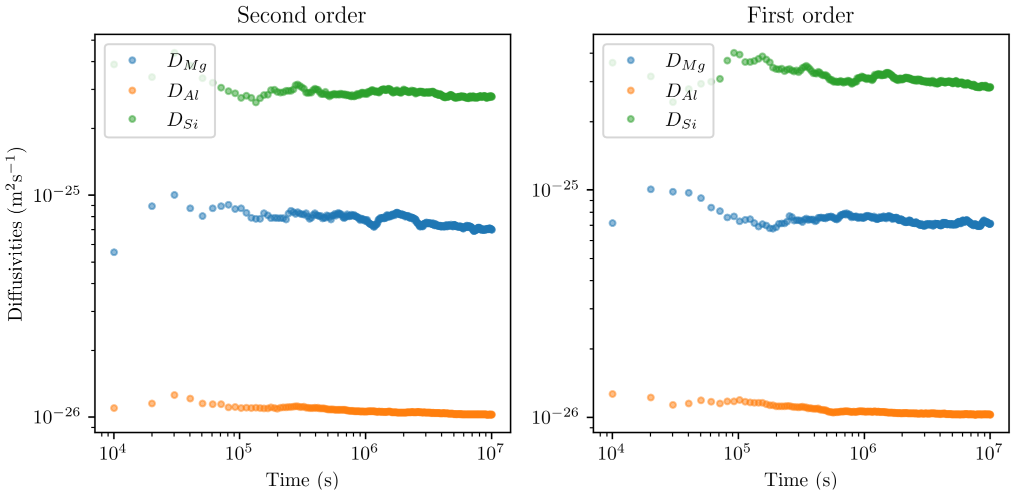


Figure 4.22: Plots of the diffusitives for Si, Mg and Al logged by KMC using first and second order RTA for the Liang method. [4]

A vacancy trapping is not purely destructive for a KMC simulation when the energy wells are large enough. If the vacancy is in an area of solutes that are contributing to lower activation energies, the trapping ensures that the vacancy does less of unproductive jumps in pure Al, and do more jumps moving solutes [48]. For the trapping with CE, however, the vacancy trapping is between two or three states, and instead of contributing to faster dynamics, the system is going back and fourth for million of iterations. This is the effect that is removed from doing the second order RTA.

4.6 Discussion of the results, sources of errors and possible solutions.

This section will take a look at the overall results and discuss their sources of errors and the possible alternatives and solutions.

4.6.1 Drawbacks of the on-lattice KMC

When using an on-lattice KMC model as in this project, the simulation is restricted to the FCC lattice. The range of events that are able to be chosen for this lattice are also predetermined, limiting other possible events like next nearest neighbours, dislocations or jump to interstitial sites. In order to evolve the system correctly using this restriction, the calculation of activation energy must be able to reflect the effects of this implicitly while still residing on the FCC lattice.

Using the on-lattice KMC for the problem also requires activation energies that accurately reflects the diffusion process in every stage of the simulation, including the beginning where the solutes are dispersed in the solid solution, when the solid starts to form small clusters, and when the precipitates are formed. This requires fine tuning of parameters with training data that accurately represents each phases, which is non-trivial to do using machine learning on one set of coefficients.

The lack of the initial clustering indicates that some part of the simulation step is missing. The activation energies or the way the activation energies are calculated using the structure in KMC must therefore lack important details of the effects that solutes in this solid solution has. The rigid FCC structure that KMC is using might be preventing key events that is initiating the effects that leads to clustering. This is not trivial problem to solve in a simulated system that is rigid to lattice position, but the implementation of octahedral sites expands the possibility to include events that are included. However, using the current dataset and cluster expansion, the configurations seen to make the jump into octahedral does not occur, so it's uncertain whether including an octahedral event using the current dataset would lead to better clustering.

An alternative to remove the problems with FCC restriction is using off-lattice KMC methods like k-ART [54]. For these methods the rates are calculated on-the-fly based on the surrounding environment and not calculated in advance as in traditional KMC methods. This off-lattice model is enabling the long-range interactions and lattice deformations that the restrictions of the on-lattice model does not have. With the on-lattice methods, these effects have to be included implicitly from the methods that are used for calculating the rates in advance, which is attempted using DFT in this work. Using this method will remove the restriction that comes with the on-lattice model by allowing events that are not represented by the FCC lattice, and by removing the limitations of only including important effects implicitly. The major drawback to this model is the increased computation time. The motivation for using the on-lattice is due to it fast computation time, and since there is interest in the long aged precipitation process in Al-Mg-Si alloys, an on-lattice KMC is to preferred if being able to represent this system. The KMC in this project is one step further by allowing octahedral jumps, but the lack of clustering using CE and KRA still indicates the need for improvement until these jumps are done by the correct physics.

4.6.2 Limitations of the training set

Using a super cell of 3×3×3 unit cell for the DFT calculation is effective for the quantitative production of datasets, however, it might lack important effects that found in real materials required to see the precipitation and clustering found in materials. A system of 108 atoms in perfect conditions is a large simplification of the real physical system. This simplification is most likely not able represent all the strain effects and long-range interactions that is seen in a material and the finer details that promotes the clustering and precipitation dynamics seen in the alloys.

A solution to check the impact of larger super cells would be to create a training set using larger cells, for example 5 × 5 × 5 unit cells. Since larger systems require more computation time, a solution would be to reduce the accuracy with the benefits of a wider range of configurations.

Currently the most of the training set consist of 1-3 solutes, as seen in table 4.1. This may result in ECIs that are mostly influenced by the effects that 1-3 solutes have on the system. Furthermore, the addition of extra vacancies doubles the amount of coefficients, increasing the coefficient-to-data ratio. Increasing the training set with solutes will reduce the probability for having ECIs that only is represented by one or two data entries, in turn reducing the risk of overfitting.

4.6.3 Errors in rate calculations

Liang rates

The methods used for calculating the activation energies comes with limitations. The Liang [4] method uses the assumption that the activation energies is mostly depending on the change in the interaction with the nearest neighbours. The interaction coefficients and the method used to calculate them is therefore responsible for the resulting behaviour during simulation. It is therefore important that also the method to obtain the coefficients accounts for the correct physics for the system at hand. The interaction coefficients used by Liang is taken from article by Hiroshawa et al.[31] where the *ab inito* calculations were done using 20 atoms. When using the new coefficients calculated from DFT in 4.5, the system no longer clusters as seen in figure 4.13. The reason for this might be the coefficients from [31] are calculated from a small cell where the bonds between pair atoms might have a large effect on the total energy. This was further confirmed by checking a second source [9] where it was used 500 atoms in the bulk calculations. The strong interaction coefficients leads to an overestimate of the repulsive and attractive effects resulting to excessive clustering.

Cluster Expansion

As mentioned in the previous paragraphs, the lack of clustering using CE could be due to lattice restrictions and lack of information in the training set. Other limitations by the CE method is the limitations by the fitting process. Machine learning processes is sensitive to overfitting when the number of coefficients are large compared to the number of entries in the dataset. Using penalized methods, like Ridge Regression used in SKMC, is an attempt at avoiding the overfitting, but the efficiency of the penalization is limited when the number of coefficients is 2 more than the number of data entries as used currently. Even though the LOOCV score of the data to the dataset is 27.14 meV, the predicted configurations encountered during the simulation might have high uncertainty. Expanding the training set to all possible configurations would be ideal, however when using computational heavy methods like DFT, this is not practical. A solution to this problem is introduced the recent article of Kleiven et al. [55]. This model ensures that the CE model is not overfitted by using a bootstrapping scheme that samples configurations from the Monte Carlo simulation. This effectively tests the ECIs against structures that are likely to occur while doing the simulation, making the model is less restricted to the ability of the training set to represent the simulation. This will in turn requiring less data and reducing the sensitivity to overfitting, which is an advantage for training sets that require larger resources like the ones needed for the activation energies used in CE.

Kinetically resolved activation barrier

The KRA method inherits the some of limitations of the CE method by using the same scheme and training set for the average barrier. Another possible error in the CE of the average barrier, is how the possible KECIs are defined. In other works using the KRA method [30][37][48], the possible KECIs are defined as the distance from the transitional state, the state between i and j, usually defined as the middle between the two sites. By defining the clusters from the distance of the jump atom as done in the case of this work, the average barrier is not linked to the distance from both, but rather the distance from the jump atom on site i. In other words, if only the configuration of the forward jump, where the jump atom is at i and vacancy at j, is in the training set, the KECIs would in reality only represent the jump from site i to site j, as the distance would be different if the jump atom was at site j and vacancy at site i. For this work most of the jumps has both the forward and reverse jump present in the training set, making both impact the KECIs which may counteract this effect. However, this may contribute to overfitting by having two equal structure twice and the ideal method would have been define the clusters from the middle of the jump and only have configuration appear once to avoid similar entries in the training set.

The additional errors that accompanies the KRA method is the errors that comes with the calculations of ΔE. To obtain it's advantage of being independent on how accurately E_{saddle} is calculated and represented by CE, the method used for finding ΔE must be found using the total energies. This introduces the problem seen by the CE of total energies found in this work. Large configurations with many atoms results in many structures to be accounted for using just few coefficients. The model performs well relative to the total energy, but lacks the finer details needed to accurately represent ΔE between two adjacent states as seen in figure 4.9, introducing errors to the activation energies. The problem of overfitting as discussed for the activation energies are not as relevant for this dataset, as the number of coefficients for this CE is drastically reduced. The accuracy may be improved by implementing triplets and quadruplets, but doing this will significantly reduce the computation time during simulation.

The reduction of drift in the total energy evolution by using KRA seen in figure 4.17, shows that the problem from the energy drift stems mostly from fit of the activation energy. This is most likely due to E_{saddle} being sensitive to the local environment as seen from the results from the BEP relation in figure 4.5. The sensitivity to the local environment might introduce uncertainties when applied to new configurations not seen in the training set, and since the total energy difference is logged as $\Delta E = E_{ji}^{act} - E_{ij}^{act}$, the logged energy is not separated from the uncertainties of E_{saddle}. Separating ΔE from the fitting procedure restricts the uncertainties in E_{saddle} to the value of the average barrier, Q, and since Q for the two adjacent states is considered the same, this uncertainty is cancelled in the logging.

Reducing the uncertainties of the E_{saddle} to new structures would be reduced if the training set was improved to represent the simulated system better using the forward activation energy. The local environments effect on the activation energy might also be different using larger supercell when calculating the dataset. Improving the training set or updating the CE fitting regime to methods like Kleiven et al. [55] would most likely also reduce the level of drift.

It is important to remember the motivation behind the KRA method. The total energy evolution is not affecting the simulation by itself, but the drift does indicate that there is a skew in the distribution function. This was presented in section 2.2 as the criterion of detailed balance, which is one of the fundamentals of doing correct sampling of the system when doing Monte Carlo simulations. Removing the energy drift does not assure that the time evolution of the system is correct however, as the main contribution of the evolution is the accuracy of the activation energies. The KRA method is currently not representing the activation energies well, which is probably a combination of both lack of structures and correct long-range effects in the training data and the CE of the total energies being too simple to represent the finer effects in the configuration.

Chapter 5

Conclusion

The objective of this master thesis was to continue from the works of Nygård [3] on the SKMC project to be able to simulate clustering and precipitation in Al-Mg-Si alloys using the kinetic Monte Carlo algorithm. This was done by attempting to solve errors introduced with the cluster expansion method previously implemented in order to get the system to cluster using physically accurate activation energies. The motivation for this is to gain insight from the important precipitation process, which is of interest for both scientific and commercial purposes.

During the work of this thesis the previously implemented cluster expansion was done using a training set with activation energies from DFT calculations. The resulting fit had an CV score of 27.14 meV and a RMSE of 9.14 meV which was a higher error than the works of Nygård using a smaller training set with less variation of structures. The kinetically resolved activation barrier was implemented and tested with the goal of correcting an energy drift occurring due to the cluster expansion method. This involved doing a cluster expansion on the average barrier for two adjacent states, which resulted in an LOOCV score of 20.11 meV and RMSE of 5.14 meV. The KRA method included the need to describe the differences in total energies between the adjacent states, which was solved by using a cluster expansion for the total energy. The cluster expansion of the total energy had a RMSE of 58.78 meV which introduced errors to the activation energies that resulted in a RMSE of 38.71 meV. KRA method was also tested using a simplified model for the total energy difference using interaction coefficients calculated from bulk calculations, and the resulting activation energy had a RMSE of 33.10 meV.

The expressions for activation energies was used in kinetic Monte Carlo simulations and the time evolution did not show clustering in room temperature using CE or KRA with the expanded training set. The energy evolution of the system was seen to obtain a large negative drift with CE but using the KRA method decreased the drift significantly. There was still some observed drift that most likely stems from the coefficients from the average barrier being defined from the jump atom and not from the transitional state.

The Liang method from article [4] was simulated both using the old interaction coefficients used in the article and updated interaction coefficients calculated from DFT bulk calculations. Using the coefficients from the article resulted in the solutes clustering to a $L1_0$ structure while new interaction coefficients resulted in no clustering. This indicates that the interaction coefficients used by Liang in the article might be overestimated due to being calculated from bulk calculation using smaller supercells.

The possibility to form β'' eyes was implemented by adding the possibility to jump into an octahedral site if the configurations were meeting a predetermined criterion. The jumps into octahedral observed during simulation does not correspond to the octahedral jumps observed in DFT calculations, and the lack of clustering prevents structures that has been observed to enable the octahedral jumps in DFT calculations to form. The implementation still lacks to do a jump based on physical activation energies, but this is enabled as a possible expansion.

The problem with vacancy trapping and system getting during simulation was solved by implementing the second order residence time algorithm. The implementation increased the time spent each KMC step 10 times for Liang method and 12 times for KRA and CE method, but the time to finish a simulation using the second order residence time algorithm was decreased to 1/3 and 1/50 of the time spent during the first order run for Liang and CE respectively. This was done by effectively eliminating 67% and 84% of the reversal jumps for the Liang and CE method.

The overall conclusions for the work is that the new implementations are a step further into describing the precipitation process of Al-Mg-Si alloys by using KMC, but the model is still lacking the important clustering that enables precipitates like β'' structures to happen. Better representation of effects in the Al-Mg-Si alloy should be further investigated as the KMC algorithm is fast and enables long time scales to be explored. Using second order RTA has improved the simulation time by making it minimum three times as fast and ensures that simulation using CE does not get stuck during the simulation. The implementation of octahedral site can easily be expanded to work with physical rates but since the system does not seem to form configurations that forms β'' in the first place, the prioritized next step should be to look at getting the KMC to represent correct effects during simulation by investigating the effects of more varied training set.

Chapter 6

Future

This section will present some of the possible future improvement on the model that could lead to better simulations with results

6.1 Better representing of CE

The current dataset calculated on a 3 × 3 × 3 might be to small to include correct information about the strain effects of certain configurations. Trying DFT calculations on larger systems will be a good approach for testing if there is missing information in the current dataset. If there is found a significant improvement, a larger dataset using calculations from larger cells should be considered. Another solution to trying to get better fit using CE is using a method like Kleiven et al. [55] using the configuration met in a MC simulation as a bootstrapping element in the procedure.

6.2 Correct formation of β"

The next step in terms of β" formation is to be able to form the precipitate under the correct conditions from DFT calculations. Currently, the model only makes a jump when configurations around the jump atom satisfy the predetermined conditions. The swap that is made, by jumping in to octahedral frees a vacancy that is now residing where the jump atom was before the jump. Currently this vacancy gets trapped, it is not included in the diffusion process and its not interacting with other vacancies. It should be found a solution that enables ergodicity ensuring that all sites can be visited, which is one of the three main requirement for a good KMC model. This means that also the octahedral site should be able to jump out.

Currently the simulation using CE or KRA never forms the configurations that is seen to enable the jump to octahedral that jumps, but when the correct foundations for these types of event is happening the jumps should happen when they are physically correct. This can probably be done several ways, but two are natural two the way the method is implemented. One solution would be to use the calculation of activation energy for the octahedral jump, making the octahedral jump be chosen according to its rate, as the rest of events in the event list. This solution needs the ability for calculating the activation energies for the octahedral jump as an isolated event, which might be hard to represent in a training set. This also requires a new cluster expansion including octahedral sites.

Having β" precipitates as a possible structure also requires that the effects from these being present in the system must be included and represented by the ECIs.

The activation energies for a jump in octahedral must be included if the jump is to be motivated by rates, and the effect of having β" in the system must be represented. This could be done by including a dataset with jumps in and out of the octahedral sites, and creating `Eact_data` for these types of jumps.

Another solution is to add a set of configurations that are seen to happen which is not chosen according to rate but would happen based on if the configuration is correct.

6.3 Improving KRA

Since Q is not fully independent of direction in the current implementation, the next step would be to define the ECIs from the configuration at transition state. The solution would be to define the transition state to be the middle between the vacancy and the jump atom. Examples in literature where this is done is the work by Zhang et al. [48].

The current implementation of the total energies do not satisfy the accuracy wanted to represent the energy differences. Fitting the total energies for a larger system than $3 \times 3 \times 3$ might give more accurate representation of the effects that solutes has on larger systems. A natural next step for activation energies is including larger clusters, like triplets or quadruplets. This will complicate the calculations done during the simulation and will increase the computation time. Adding prior functions to the Bayesian regression as done with the local cluster expansion would also be a natural next step in order to tune the fitting to be more based on physical intuition.

Bibliography

[1] J. Hirsch, 'Automotive trends in aluminium-the european perspective,' in *Materials Forum*, vol. 28, 2004, pp. 15–23.

[2] Ø. Ryen, B. Holmedal, O. Nijs, E. Nes, E. Sjölander and H.-E. Ekström, 'Strengthening mechanisms in solid solution aluminum alloys,' *Metallurgical and Materials Transactions A*, vol. 37, no. 6, pp. 1999–2006, 2006.

[3] Ø. T. Nygård, 'Kinetic monte carlo simulations of the early stages of precipitation in al-mg-si alloys using ab initio based activation energies,' M.S. thesis, Norwegian University of Science and Technology, 2020.

[4] Z. Liang, C. S. T. Chang, C. Abromeit, J. Banhart and J. Hirsch, 'The kinetics of clustering in al–mg–si alloys studied by monte carlo simulation,' *International journal of materials research*, vol. 103, no. 8, pp. 980–986, 2012.

[5] R. Lumley, *Fundamentals of aluminium metallurgy: recent advances.* Woodhead Publishing, 2018.

[6] G. Mrówka-Nowotnik, 'Influence of chemical composition variation and heat treatment on microstructure and mechanical properties of 6xxx alloys,' *Archives of materials science and engineering*, vol. 46, no. 2, pp. 98–107, 2010.

[7] R. Garrett, J. Lin and T. Dean, 'An investigation of the effects of solution heat treatment on mechanical properties for aa 6xxx alloys: Experimentation and modelling,' *international Journal of Plasticity*, vol. 21, no. 8, pp. 1640–1657, 2005.

[8] M. Kubota, J. F. Nie and B. C. Muddle, 'Characterisation of precipitation hardening response and as-quenched microstructures in al-mg (-ag) alloys,' *Materials transactions*, vol. 45, no. 12, pp. 3256–3263, 2004.

[9] E. A. Mortsell, S. J. Andersen, J. Friis, C. D. Marioara and R. Holmestad, 'Atomistic details of precipitates in lean al–mg–si alloys with trace additions of ag and ge studied by haadf-stem and dft,' *Philosophical Magazine*, vol. 97, no. 11, pp. 851–866, 2017.

[10] C. D. Marioara, S. Andersen, H. Zandbergen and R. Holmestad, 'The influence of alloy composition on precipitates of the al-mg-si system,' *Metallurgical and materials transactions A*, vol. 36, no. 3, pp. 691–702, 2005.

[11] P. H. Ninive, A. Strandlie, S. Gulbrandsen-Dahl, W. Lefebvre, C. D. Marioara, S. J. Andersen, J. Friis, R. Holmestad and O. M. Lovvik, 'Detailed atomistic insight into the β "phase in al–mg–si alloys,' *Acta materialia*, vol. 69, pp. 126–134, 2014.

[12] C. Marioara, S. Andersen, J. Jansen and H. Zandbergen, 'The influence of temperature and storage time at rt on nucleation of the beta phase in a 6082 al–mg–si alloy,' *Acta Materialia*, vol. 51, no. 3, pp. 789–796, 2003.

[13] G. Meyruey, V. Massardier, W. Lefebvre and M. Perez, 'Over-ageing of an al-mg-si alloy with silicon excess,' *Materials Science and Engineering: A*, vol. 730, pp. 92–105, 2018.

[14] N. Metropolis, 'The beginning,' *Los Alamos Science*, vol. 15, pp. 125–130, 1987.

[15] K. M. Thompson, D. E. Burmaster and E. A. Crouch3, 'Monte carlo techniques for quantitative uncertainty analysis in public health risk assessments,' *Risk Analysis*, vol. 12, no. 1, pp. 53–63, 1992.

[16] A. Bouland, W. van Dam, H. Joorati, I. Kerenidis and A. Prakash, 'Prospects and challenges of quantum finance,' *arXiv preprint arXiv:2011.06492*, 2020.

[17] M. N. Rosenbluth, 'Genesis of the monte carlo algorithm for statistical mechanics,' in *AIP Conference Proceedings*, American Institute of Physics, vol. 690, 2003, pp. 22–30.

[18] N. Metropolis and S. Ulam, 'The monte carlo method,' *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[19] R. L. Andrew, 'Molecular modeling principles and applications,' *2nd, editor.: Pearson Education Limited*, 2001.

[20] A. F. Voter, 'Introduction to the kinetic monte carlo method,' in *Radiation effects in solids*, Springer, 2007, pp. 1–23.

[21] M. Newman and G. Barkema, *Monte carlo methods in statistical physics chapter 1-4*. Oxford University Press: New York, USA, 1999, vol. 24.

[22] A. B. Bortz, M. H. Kalos and J. L. Lebowitz, 'A new algorithm for monte carlo simulation of ising spin systems,' *Journal of Computational Physics*, vol. 17, no. 1, pp. 10–18, 1975.

[23] K. M. Carling, G. Wahnström, T. R. Mattsson, N. Sandberg and G. Grimvall, 'Vacancy concentration in al from combined first-principles and model potential calculations,' *Physical Review B*, vol. 67, no. 5, p. 054 101, 2003.

[24] M. Athenes, P. Bellon and G. Martin, 'Identification of novel diffusion cycles in b2 ordered phases by monte carlo simulation,' *Philosophical Magazine A*, vol. 76, no. 3, pp. 565–585, 1997.

[25] T. Rautiainen and A. Sutton, 'Influence of the atomic diffusion mechanism on morphologies, kinetics, and the mechanisms of coarsening during phase separation,' *Physical Review B*, vol. 59, no. 21, p. 13 681, 1999.

[26] C. Daniels and P. Bellon, 'Hybrid kinetic monte carlo algorithm for strongly trapping alloy systems,' *Computational Materials Science*, vol. 173, p. 109 386, 2020.

[27] D. Mason, R. Rudd and A. Sutton, 'Stochastic kinetic monte carlo algorithms for long-range hamiltonians,' *Computer physics communications*, vol. 160, no. 2, pp. 140–157, 2004.

[28] J. M. Sanchez, F. Ducastelle and D. Gratias, 'Generalized cluster description of multicomponent systems,' *Physica A: Statistical Mechanics and its Applications*, vol. 128, no. 1-2, pp. 334–350, 1984.

[29] T. Mueller and G. Ceder, 'Bayesian approach to cluster expansions,' *Physical Review B*, vol. 80, no. 2, p. 024 103, 2009.

[30] A. Van der Ven, G. Ceder, M. Asta and P. Tepesch, 'First-principles theory of ionic diffusion with nondilute carriers,' *Physical Review B*, vol. 64, no. 18, p. 184 307, 2001.

[31] S. Hirosawa, F. Nakamura and T. Sato, 'First-principles calculation of interaction energies between solutes and/or vacancies for predicting atomistic behaviors of microalloying elements in aluminum alloys,' in *Materials Science Forum*, Trans Tech Publ, vol. 561, 2007, pp. 283–286.

[32] D. B. Laks, L. Ferreira, S. Froyen and A. Zunger, 'Efficient cluster expansion for substitutional systems,' *Physical Review B*, vol. 46, no. 19, p. 12 587, 1992.

[33] R. Magri and A. Zunger, 'Real-space description of semiconducting band gaps in substitutional systems,' *Physical Review B*, vol. 44, no. 16, p. 8672, 1991.

[34] H. Geng, M. Sluiter and N. Chen, 'Cluster expansion of electronic excitations: Application to fcc ni–al alloys,' *The Journal of chemical physics*, vol. 122, no. 21, p. 214 706, 2005.

[35] C. Ravi, H. Sahu, M. Valsakumar and A. van de Walle, 'Cluster expansion monte carlo study of phase stability of vanadium nitrides,' *Physical Review B*, vol. 81, no. 10, p. 104 111, 2010.

[36] A. Seko, K. Yuge, F. Oba, A. Kuwabara, I. Tanaka and T. Yamamoto, 'First-principles study of cation disordering in mg al 2 o 4 spinel with cluster expansion and monte carlo simulation,' *Physical Review B*, vol. 73, no. 9, p. 094 116, 2006.

[37] X. Zhang and M. H. Sluiter, ‘Cluster expansions for thermodynamics and kinetics of multicomponent alloys,’ *Journal of Phase Equilibria and Diffusion*, vol. 37, no. 1, pp. 44–52, 2016.

[38] N. A. Zarkevich and D. D. Johnson, ‘Reliable first-principles alloy thermodynamics via truncated cluster expansions,’ *Physical review letters*, vol. 92, no. 25, p. 255 702, 2004.

[39] G. Henkelman and H. Jónsson, ‘Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points,’ *The Journal of chemical physics*, vol. 113, no. 22, pp. 9978–9985, 2000.

[40] G. Henkelman and H. Jónsson, ‘A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives,’ *The Journal of chemical physics*, vol. 111, no. 15, pp. 7010–7022, 1999.

[41] A. E. Hoerl and R. W. Kennard, ‘Ridge regression: Biased estimation for nonorthogonal problems,’ *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[42] M. W. Browne, ‘Cross-validation methods,’ *Journal of mathematical psychology*, vol. 44, no. 1, pp. 108–132, 2000.

[43] Y. Wang, J. Yin, X. Liu, R. Wang, H. Hou and J. Wang, ‘Precipitation kinetics in binary fe–cu and ternary fe–cu–ni alloys via kmc method,’ *Progress in Natural Science: Materials International*, vol. 27, no. 4, pp. 460–466, 2017.

[44] H. Ko, J. Deng, I. Szlufarska and D. Morgan, ‘Ag diffusion in sic high-energy grain boundaries: Kinetic monte carlo study with first-principle calculations,’ *Computational Materials Science*, vol. 121, pp. 248–257, 2016.

[45] J. Bronsted, ‘Acid and basic catalysis.,’ *Chemical Reviews*, vol. 5, no. 3, pp. 231–338, 1928.

[46] M. Evans and M. Polanyi, ‘Inertia and driving force of chemical reactions,’ *Transactions of the Faraday Society*, vol. 34, pp. 11–24, 1938.

[47] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh and C. Fiolhais, ‘Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation,’ *Physical review B*, vol. 46, no. 11, p. 6671, 1992.

[48] X. Zhang and M. H. Sluiter, ‘Kinetically driven ordering in phase separating alloys,’ *Physical Review Materials*, vol. 3, no. 9, p. 095 601, 2019.

[49] J. H. Chang, D. Kleiven, M. Melander, J. Akola, J. M. Garcia-Lastra and T. Vegge, ‘Clease: A versatile and user-friendly implementation of cluster expansion method,’ *Journal of Physics: Condensed Matter*, vol. 31, no. 32, p. 325 901, 2019.

[50] M. Själander, M. Jahre, G. Tufte and N. Reissmann, *EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure*, 2019. arXiv: `1912.05848 [cs.DC]`.

[51] D. Kleiven and J. Akola, 'Precipitate formation in aluminium alloys: Multi-scale modelling approach,' *Acta Materialia*, vol. 195, pp. 123–131, 2020.

[52] M. Mantina, Y. Wang, L. Chen, Z. Liu and C. Wolverton, 'First principles impurity diffusion coefficients,' *Acta Materialia*, vol. 57, no. 14, pp. 4102–4108, 2009.

[53] O. M. Løvvik, E. Sagvolden and Y. Li, 'Prediction of solute diffusivity in al assisted by first-principles molecular dynamics,' *Journal of Physics: Condensed Matter*, vol. 26, no. 2, p. 025 403, 2013.

[54] N. Mousseau, L. K. Béland, P. Brommer, F. El-Mellouhi, J.-F. Joly, G. K. N'Tsouaglo, O. Restrepo and M. Trochet, 'Following atomistic kinetics on experimental timescales with the kinetic activation–relaxation technique,' *Computational Materials Science*, vol. 100, pp. 111–123, 2015.

[55] D. Kleiven, J. Akola, A. A. Peterson, T. Vegge and J. H. Chang, 'Training sets based on uncertainty estimates in the cluster-expansion method,' *Journal of Physics: Energy*, vol. 3, no. 3, p. 034 012, 2021.

NTNU

Norwegian University of
Science and Technology