Long-timescale atomistic simulations

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Johannes Bulin
aus
Dresden

Bonn, 2020

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter:	Prof. Dr. Michael Griebel
2. Gutachter:	Prof. Dr. Marc Alexander Schweitzer
Tag der Promotion:	11.03.2021
Erscheinungsjahr:	2021

Acknowledgements

I would like to thank everyone who made it possible for me to write this thesis. I am grateful for all the time that James Barker, Astrid Maaß, Christian Neuen, Jutta Neuen, and Tobias Olbrich have spent finding typos, missing indices as well as half-finished sentences. In particular, I would also like to thank Prof. Dr. Michael Griebel and Dr. Jan Hamaekers who introduced me to this topic, gave me the freedom to experiment and provided guidance. Finally, I am grateful for all the support from my family and friends who backed me and put up with me.

Contents

1 Introduction 1

2 Atomistic simulations 5
- 2.1 From quantum mechanics to classical mechanics . . . 5
- 2.2 The ensemble concept . . . 8
- 2.3 Observables . . . 10
- 2.4 Molecular dynamics . . . 11
- 2.5 Metastability and long-timescale simulations . . . 13

3 Coarse-grained long-timescale simulations 15
- 3.1 Discretising the position space . . . 15
  - 3.1.1 Transition state theory . . . 17
  - 3.1.2 Harmonic transition state theory . . . 18
  - 3.1.3 Other theories . . . 19
- 3.2 The Kinetic Monte Carlo algorithm . . . 19
- 3.3 On-the-fly Kinetic Monte Carlo algorithms . . . 20
- 3.4 Accelerated Dynamics Methods . . . 21

4 Localised on-the-fly Kinetic Monte Carlo algorithms 23
- 4.1 Basic localisation theory . . . 24
- 4.2 Localised on-the-fly Kinetic Monte Carlo algorithms . . . 27
- 4.3 Comparison functions for atomic neighbourhoods . . . 30
- 4.4 The invariant RMSD comparison function . . . 33
- 4.5 Calculating the invariant RMSD . . . 35
  - 4.5.1 The case $\epsilon = 0$ . . . 39
  - 4.5.2 The case $\epsilon > 0$ . . . 40
  - 4.5.3 Computational complexity . . . 47
- 4.6 The local transition database . . . 54
  - 4.6.1 Database structure . . . 55
  - 4.6.2 Storing local transitions . . . 57
  - 4.6.3 Retrieving local transitions . . . 57

4.7 The Localised Kinetic Monte Carlo algorithm . . . 58
4.7.1 Local transition searches . . . 59
4.7.2 Confidence estimation . . . 61
4.7.3 Recycling local transitions . . . 61

5 Evaluation **63**
5.1 Example problems . . . 63
5.1.1 Ad-atom hopping on a platinum surface . . . 63
5.1.2 Hydrogen diffusion in amorphous silicon . . . 64
5.1.3 Nickel grain boundary . . . 64
5.2 Localisation of transitions . . . 65
5.3 Comparing atomic neighbourhoods . . . 68
5.3.1 Accuracy . . . 70
5.3.2 Performance evaluation . . . 71
5.3.3 Scaling with respect to n . . . 74
5.3.4 Scaling with respect to ϵ . . . 76
5.3.5 Testing different invariances . . . 78
5.4 LKMC . . . 82
5.4.1 Platinum ad-atom . . . 83
5.4.2 Hydrogen diffusion in amorphous silicon . . . 85
5.4.3 Nickel grain boundary . . . 87

6 Conclusions **89**

A Appendix **91**
A.1 Fixing Go-Permdist . . . 91
A.2 Simulation parameters . . . 91

Bibliography **93**

List of Figures **101**

List of Tables **103**

List of Algorithms **105**

CHAPTER 1

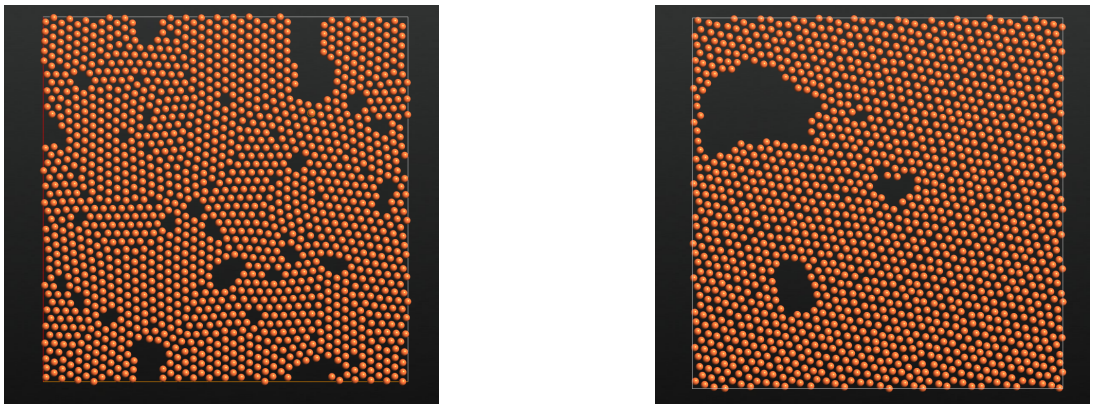
Introduction

Historically, materials science was basically limited to trial-and-error style experiments. Known substances were mixed, heated, or otherwise transformed, and their macroscopic properties analysed. Today far more advanced techniques are available to investigate materials, their physical properties, and their reaction behaviour.

Besides growing physical and chemical knowledge and the increasing accuracy of experiments, computational methods offer new insights into the behaviour of materials. They can – if used correctly – provide information where classical experiments fail, for example when it is not possible to detect certain properties without disturbing the system under observation or where the temporal or spatial scales become too small. They can also be faster or cheaper than classical experiments which may require expensive equipment as well as trained scientists. A wide range of computational methods are available, depending on the properties of interest. They range from stress simulations for whole buildings to quantum mechanical calculations in order to determine the properties of single atoms.

This thesis deals with the problem of simulating the dynamics of atomistic systems that are in thermal contact with a heat bath. If timescales of more than a few microseconds are required, the standard Molecular Dynamics (MD) approach can no longer be used – thus other algorithms are needed to predict the movement of atoms over such periods of time. Reaching long timescales is important in a number of interesting simulation problems – both in academia and in industry. One such application is the simulation of ageing processes in materials. On an atomistic level, ageing can happen in multiple ways: defects that are present in a material can move and merge, forming larger defects or microcracks as shown in figure 1.1. Certain impurities may also diffuse through the material, causing changes to the original material structure. The latter is a major problem in nuclear power plants, where the diffusion of hydrogen causes the reactor vessel to become brittle [1, 2], leading to potentially catastrophic consequences. As the word *ageing* implies, it is not sufficient to simulate only a few microseconds of such processes. Instead, reaching timescales of at least hours is crucial in order to obtain reasonable results.

Another area of interest is the investigation of deposition processes. Different kinds of



(a) Initial system – many small defects

(b) Final system – few large defects

Figure 1.1: Formation of larger defects from smaller ones.

deposition processes are used to create high-purity silicon wafers in the semiconductor industry. Depending on the intended use, various types of silicon are required, ranging from perfect silicon crystals through polycrystalline silicon (as in figure 1.2) to amorphous silicon. Varying the deposition method, deposition rate, or the ambient conditions may result in very different crystal structures. Therefore understanding the influence of these parameters is crucial for optimising the manufacturing process. As deposition processes may progress slowly – growth rates of only slightly more than $1\,\mathrm{nm\,min^{-1}}$ are quite common [3] – it is once again necessary to simulate timescales of at least a couple of seconds.

Finally, such simulations also play an important role in computational biology. To understand certain biological processes it is important to understand how some molecules – specifically proteins – evolve over time. Some proteins fold, meaning that their structure changes from a chain-like structure to a denser (folded) three-dimensional packing. This folding process may take from a few microseconds up to several hours [4], depending on the protein. The folded and the unfolded states of a protein may have different biological properties, making the questions of how, why, and when they fold an important problem in biology.

Various approaches to tackle the timescale problem exist. In this thesis, the focus is on on-the-fly Kinetic Monte Carlo algorithms that exploit inherent metastability in particle systems to simulate their dynamics on a coarser scale. As their name suggests, such algorithms search continuously for new metastable states in a particle system, which can be computationally prohibitive. One possible amendment to this problem was pioneered in the *kinetic activation relaxation technique* (k-ART). There, information about local structural changes that can occur in a particle system are stored and recycled whenever possible in order to find new metastable states. Due to the design of k-ART, this recycling step can fail if the particle system is amorphous.

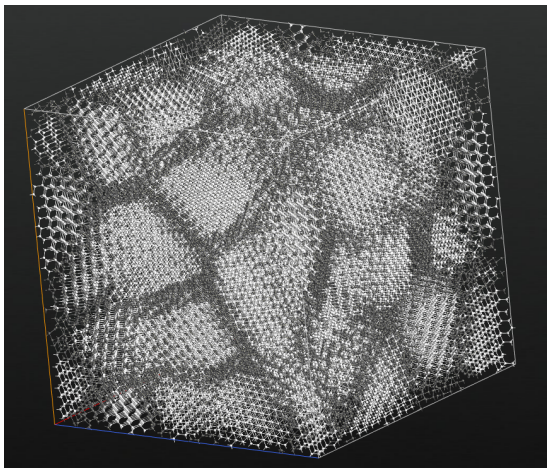


Figure 1.2: Polycrystalline silicon - grain boundaries are dark grey.

Therefore, I introduce a modification of k-ART, called the *Localised Kinetic Monte Carlo algorithm* (LKMC) that eliminates some of the problems of the classical k-ART algorithm. The centrepiece is a new approach for the comparison of particle systems that is invariant under both permutations and orthogonal transformations. It is based on a previously known variant of the root mean square deviation (RMSD) that possesses the same invariant properties. Calculating this invariant RMSD is in general NP-hard. However, I show that in the context of k-ART this invariant RMSD can be exactly calculated in polynomial time with respect to the number of particles. A corresponding algorithm is presented that can not only compare particle systems but is also able to calculate the optimal permutation and orthogonal transformation that maps two particle systems onto each other. This comparison procedure is then used in the LKMC algorithm and applied to a number of test problems where it shows promising accuracy for amorphous particle systems.

The basics of particle system dynamics are presented in chapter 2, followed by an overview of long-timescale algorithms in chapter 3. The k-ART algorithm is introduced in chapter 4, as well as the LKMC algorithm and all its components. The LKMC algorithm is then applied to a number of test problems in chapter 5 and its behaviour is analysed. Finally, the results are summarised in chapter 6 and additional thoughts related to this work are presented.

CHAPTER 2

Atomistic simulations

Before the problem of long-timescale atomistic simulations is tackled, more fundamental questions have to be answered. Questions like: How do atoms move? How do they interact with each other? And why do we need long atomistic simulations in the first place?

Answers to these questions are presented in this chapter. The very basics of atom movements can be found in section 2.1 while the central question *Why do we need long atomistic simulations* is answered in section 2.5.

2.1 From quantum mechanics to classical mechanics

Ignoring relativistic effects and assuming that electrons and atomic nuclei are stable, quantum mechanics (QM) offer currently the most accurate description of their behaviour [5]. Treating atoms in a QM setting introduces a variety of challenges, the largest one being that it is usually impossible to solve the necessary equations, either numerically or analytically.

Two central equations in non-relativistic QM are the time-dependent Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}\Psi(r,t) = H\Psi(r,t) \tag{2.1}$$

and its time-independent version

$$H\Psi(r) = E\Psi(r). \tag{2.2}$$

Equation (2.1) can be used to determine the *wave function* $\Psi : \mathbb{R}^{N\times 3} \times \mathbb{R} \to \mathbb{C}$ which contains all quantum mechanical information about the electrons and atomic nuclei in a system at a given time. The three-dimensional coordinates of the involved atomic nuclei and electrons are given by $r \in \mathbb{R}^{N\times 3}$, H is the *quantum Hamiltonian operator* which includes the – usually electrostatic – interactions between the electrons and nuclei and $\hbar$ is the reduced Planck constant.

Equations (2.1) and (2.2) are $3N$-dimensional complex valued partial differential equations where N is not the number of atoms but the number of electrons and atomic nuclei combined.

Due to their high dimensionality these equations are usually not numerically solvable. For very simple cases analytical solutions can be computed: equation (2.2) for example can be solved exactly for the hydrogen atom if only the single electron of the atom is treated quantum mechanically [6]. For larger systems exact solutions are not possible. Instead, other techniques can be used to calculate approximate solutions. For example, the Hartree-Fock ansatz or the density functional theory (DFT) can be used to approximate some of the eigenvalues E in equation (2.2).

In a number of applications of interest, however, it is sufficient to treat atoms as *classical particles*. Here, classical means that each atom is modelled as a point mass which moves according to Newton's laws. This simplification is usually motivated by the fact that atomic nuclei have significantly more mass than electrons. Therefore it is possible to separate the movement of the atomic nuclei – which move much more slowly than the electrons – from the movement of the electrons. Using this, the classical equations of motions for the atomic nuclei can be derived. A detailed derivation of the classical equations of motions starting from the Schrödinger equation can be found in [7].

Given a collection of n atoms that are completely isolated from any external influences, the evolution of the atomic coordinates $q(t) : \mathbb{R} \to \mathbb{R}^{n\times 3}$ and velocities $v(t) : \mathbb{R} \to \mathbb{R}^{n\times 3}$ can be simulated using the classical equations of motion

$$\begin{aligned} \dot{q}(t) &= v(t) \\ M\dot{v}(t) &= F(q(t), t). \end{aligned}$$

In this equation, $M \in \mathbb{R}^{n\times n}$ is a diagonal matrix such that $M_{i,i}$ is the mass of the i-th particle whereas $F : \mathbb{R}^{n\times 3} \times \mathbb{R} \to \mathbb{R}^{n\times 3}$ denotes the forces that act on the particles at time t. By solving this ordinary differential equation it is possible to simulate the movement of the atoms – and therefore the molecules and crystals that they form – over time.

If the atoms are completely isolated from any external influences, the forces depend only on the current positions of all atoms and do not depend explicitly on the time. In this case, the forces that act on the particles are implicitly given by a *potential energy function* $E : \mathbb{R}^{n\times 3} \to \mathbb{R}$, also known as the *potential energy surface*. This function maps from the atomic coordinates to the corresponding potential energy; in this case, the forces are related to E by

$$F(q(t), t) = -\nabla E(q(t)).$$

Using given initial atom positions $q(0)$ and velocities $v(0)$, the atom positions at time t can be determined by solving the differential equation:

$$\begin{aligned} \dot{q}(t) &= v(t) \\ M\dot{v}(t) &= -\nabla E(q(t)). \end{aligned} \tag{2.3}$$

In the context of atomistic simulations, equation (2.3) is often replaced by the equivalent

Hamiltonian formulation of classical mechanics [8]

$$\begin{aligned} \dot{q}(t) &= \frac{\partial H(q(t), p(t))}{\partial p} \\ \dot{p}(t) &= \frac{\partial H(q(t), p(t))}{\partial q}. \end{aligned} \tag{2.4}$$

Here $q(t)$ denotes as before the atom positions at time t whereas $p(t)$ stands for the momenta of the particles which are related to the velocities as $p(t) = Mv(t)$. The (classical) *Hamiltonian* $H(q, p) : \mathbb{R}^{n\times 3} \times \mathbb{R}^{n\times 3} \to \mathbb{R}$ is defined as

$$H(q, p) = E(q) + E_{kin}(p)$$

where E_{kin} is the kinetic energy of the particles, given by the formula

$$E_{kin}(p) = \sum_{i=1}^{n} \frac{\|p_{i,*}\|_2^2}{2M_{i,i}}.$$

By using the relation $p(t) = Mv(t)$ and the fact that the Hamiltonian fulfils

$$\begin{aligned} \dot{q}(t) &= \frac{\partial H(q(t), p(t))}{\partial p} = M^{-1}p(t) = v(t) \\ Mv(t) = \dot{p}(t) &= \frac{\partial H(q(t), p(t))}{\partial q} = -\nabla E(q(t)) \end{aligned}$$

one can verify that equations (2.3) and (2.4) are indeed equivalent.

One of the advantages of the Hamiltonian formulation is the connection between the Hamiltonian and the potential energy E, kinetic energy E_{kin}, and total energy $E + E_{kin}$ of a many-particle system. Given particle coordinates $q \in \mathbb{R}^{n\times 3}$ and corresponding momenta $p \in \mathbb{R}^{n\times 3}$, the Hamiltonian $H(q, p)$ represents the total energy of the system. Furthermore, for a trajectory $(q(t), v(t))$ that fulfils equation (2.4), the Hamiltonian fulfils

$$\frac{dH}{dt}(q(t), p(t)) = 0$$

and thus offers a mathematical representation of energy conservation.

The process of solving the Hamiltonian equations of motions will from now on be called *Hamiltonian Dynamics (HD)*. HD is the foundation for many atomistic materials simulations, some of which will be explained later on. There are a couple of issues to be kept in mind when using HD. One big problem is the correct choice of the potential energy function E. A huge variety of such functions are available in today's literature, ranging from simple quadratic functions to highly complex ones that stem from QM. Picking a potential energy function that is both accurate enough to reproduce the quantities of interest *and* cheap enough in terms of

computational cost remains a hard problem. In this thesis I will always assume that the chosen potential energy function is sufficiently accurate for the problem under investigation.

To complicate things even further, discretising equation (2.4) usually requires timesteps that must be as low as a femtosecond in order to avoid instabilities [9]. As a consequence, solving this equation for values of t larger than a few microseconds is usually not possible.

2.2 The ensemble concept

Using HD to calculate single trajectories is a useful tool for some applications, for example to simulate the trajectories of celestial bodies, whose motions can also be predicted using Newtonian dynamics[1]. Another application is the qualitative investigation of some reaction mechanisms. From the trajectories of the involved atoms or molecules one can draw conclusions as to how some chemical reactions take place on an atomistic level or why some reactions occur or do not occur under given conditions.

Unfortunately, analysing single HD trajectories is not sufficient for a large number of simulation problems due to a number of reasons. First, in most applications exact initial conditions for the equations of motion are not available. Even if they are known, solving these equations numerically with sufficient accuracy may be impossible, as the equations of motion can exhibit chaotic behaviour, causing an exponential growth of the error over time [7].

Second, HD in its pure form can only be used to model isolated systems. In many cases, at least some degree of interaction with other systems is desired. As even small amounts of matter may contain huge numbers of atoms, it is generally not possible to model all interacting systems on an atomistic level[2].

Solutions to these problems can be found in the vast field of *statistical mechanics* (sometimes called statistical physics). One of the central questions of statistical mechanics is how certain atomistic (or microscopic) structures are related to macroscopic properties like temperature or pressure. A fundamental concept of statistical mechanics are *ensembles*. Following the work in [8], the ensemble concept is introduced now. To begin with, a few definitions are needed.

Definition 1. *The tuple* $(q, p) \in \mathbb{R}^{n\times 3} \times \mathbb{R}^{n\times 3}$ *is called the* microscopic state *of an n-particle system. It describes the n three-dimensional positions and momenta of an n-particle system. The notation* $q_i \in \mathbb{R}^3$ *is used to denote the position of the i-th particle, and* $p_i \in \mathbb{R}^3$ *its momentum.*

Definition 2. *The* phase space Ω_n *of n-particle systems is the space of all possible microscopic states, i.e.* $\Omega_n = \mathbb{R}^{n\times 3} \times \mathbb{R}^{n\times 3}$.

For a fixed number of particles, the phase space contains all possible microscopic states. It also includes a number of microscopic states that can never occur even under the most extreme circumstances – for example systems where all particles have the same position. An ensemble

[1] The forces in this case are given by gravitational interactions.

[2] Under very optimistic conditions, specialised parallel codes can cope with up to 20 trillion atoms as of 2019 [10].

is now defined as a special probability density on the phase space. Depending on the ensemble, this may exclude these "impossible" microscopic states.

Definition 3. *An* ensemble *on the phase space Ω_n is a time-dependent probability density $\rho(q,p,t) : \Omega_n \times \mathbb{R} \to \mathbb{R}$ that fulfils the* Liouville equation

$$0 = \frac{\partial \rho(q,p,t)}{\partial t} + \sum_{i=1}^{n} \frac{\partial \rho(q,p,t)}{\partial q_i}\frac{\partial H(q,p,t)}{\partial p_i} - \frac{\partial \rho(q,p,t)}{\partial p_i}\frac{\partial H(q,p,t)}{\partial q_i} \tag{2.5}$$

for all $(p,q,t) \in \Omega_n \times \mathbb{R}$.

The Liouville equation ensures that the probability density evolves according to the HD of the microscopic states. Especially interesting are so-called *equilibrium ensembles*.

Definition 4. *An ensemble $\rho(q,p,t)$ is called an* equilibrium ensemble *if*

$$\frac{\partial \rho(q,p,t)}{\partial t} = 0$$

for all t. In this case the explicit time dependency of ρ is dropped and $\rho(q,p)$ is used instead of $\rho(q,p,t)$.

Certain equilibrium ensembles provide a connection between macroscopic properties like temperature and energy and the underlying microscopic systems. These ensembles can usually be derived directly from the basic thermodynamic laws.

The first and most basic equilibrium ensemble is the *microcanonical ensemble*, also called NVE ensemble. It is used to provide a link between microscopic states in an isolated macroscopic system with a fixed number of particles (N), a fixed volume (V) and a fixed total energy (E). If these three properties are known, the corresponding microscopic systems are distributed in phase space according to the microcanonical distribution

$$\rho^{NVE}(q,p) = \frac{1}{Z}\delta(H(q,p) - E).$$

Here, δ is the Dirac δ-distribution and Z is a normalisation constant[3]. As mentioned before, the distribution can be derived using the laws of thermodynamics (see [8]).

The microcanonical ensemble covers isolated systems that do not interact with their surroundings. In contrast, the *canonical ensemble* (also called the NVT ensemble) describes systems with a fixed number of particles (N) and a fixed volume (V) that is in thermal equilibrium with a heat bath of some temperature (T). In this context, a heat bath is an infinitely large system with a fixed temperature that may exchange heat with the system under investigation without interacting directly with its atoms.

[3] Z is also known as the partition function as it depends implicitly on N, V and E.

Similar to the microcanonical ensemble, the canonical ensemble can be derived from the laws of thermodynamics. Its density is explicitly given by

$$\rho^{NVT}(q,p) = \frac{1}{Z}\exp(-H(q,p)/(k_B T))$$

where Z is once again a normalisation constant such that ρ^{NVT} is a probability density and k_B is the *Boltzmann constant*.

Besides the microcanonical and canonical ensemble, other equilibrium ensembles are used frequently. One of them is the *isobaric-isothermal ensemble* (NPT) and the *grand canonical ensemble* (μVT) where the number of particles is no longer constant.

2.3 Observables

Now that the step from single HD trajectories to whole statistical ensembles has been taken, it is time to extract information from these equilibrium ensembles. In an ensemble context, one is usually interested in the properties of the whole ensemble, not of individual microscopic states. Given an *observation function* $a(q,p)$ on the phase space Ω_n, the corresponding *equilibrium observable* $\langle a|\rho\rangle$ for an equilibrium ensemble ρ is defined by

$$\langle a|\rho\rangle = \int_{\Omega_n} a(q,p)\, d\rho(q,p). \tag{2.6}$$

One simple example would be the average total energy of an ensemble, which is given by

$$\langle H|\rho\rangle = \int_{\Omega_n} H(q,p)\, d\rho(q,p).$$

In the case of the canonical ensemble, the calculation of ensemble averages can be simplified if the observation function depends only on the particle coordinates and not on the momenta (and can thus be written as $a(q)$). This is the case in many applications, for example when calculating average potential energies, diffusion constants or stress. As the canonical density can be split into a position and a momenta depending part, such averages can be calculated as

follows

$$\begin{aligned}
\langle a|\rho^{NVT}\rangle &= \frac{1}{Z}\left[\int_{\mathbb{R}^{n\times 3}} \exp(-E_{kin}(p)/(k_B T))\,dp\right]\left[\int_{\mathbb{R}^{n\times 3}} \exp(-E(q)/(k_B T))a(q)\,dq\right] \\
&= \frac{1}{Z}\left[\int_{\mathbb{R}^{n\times 3}} \prod_{i=1}^{n} \exp\left(-\frac{\|p_i\|_2^2}{2m_i k_B T}\right)dp\right]\left[\int_{\mathbb{R}^{n\times 3}} \exp(-E(q)/(k_B T))a(q)\,dq\right] \\
&= \underbrace{\frac{1}{Z}\left[\prod_{i=1}^{n}\sqrt{(2\pi m_i k_B T)^3}\right]}_{=:\tilde{Z}^{-1}}\left[\int_{\mathbb{R}^{n\times 3}} \exp(-E(q)/(k_B T))a(q)\,dq\right] \\
&= \int_{\mathbb{R}^{n\times 3}} a(q)\,\underbrace{\frac{1}{\tilde{Z}}\exp(-E(q)/(k_B T))}_{=:\rho_{pos}^{NVT}(q)}\,dq. \qquad (2.7)
\end{aligned}$$

Definition 5. *Similar to the definition of the phase space, the* position space Θ_n *is the space of all possible particle positions for an n-particle system, i.e.* $\Theta_n = \mathbb{R}^{n\times 3}$.

Definition 6. *The probability density*

$$\rho_{pos}^{NVT}(q) = \frac{1}{\tilde{Z}}\exp(-E(q)/(k_B T)) \qquad (2.8)$$

that occurred in equation (2.7) *will be called the canonical density in position space.*

2.4 Molecular dynamics

In order to calculate ensemble averages it is necessary to integrate a high dimensional function as shown in equation (2.6). Classical quadrature rules are usually not suited for this task, unless the number of particles is low.

While various Monte Carlo based approaches can be used to calculate ensemble averages, the focus here will be on Molecular Dynamics (MD) methods. The idea of Molecular Dynamics is to replace the ensemble average by a time average of some trajectory. In the case of the microcanonical ensemble, some equilibrium observable $\langle a|\rho^{NVE}\rangle$ is calculated by using a long HD trajectory $(q(t), v(t))$ and the relation

$$\langle a|\rho^{NVE}\rangle = \lim_{t\to\infty}\int_0^t a(q(\tau), p(\tau))\,d\tau. \qquad (2.9)$$

The assumption that the equality in (2.9) holds is called the *ergodic hypothesis* [7]. In general this is not true, however, it is often assumed to be the case [11, 12].

The MD approach for the microcanonical ensemble can be extended to cover other ensembles as well. The key idea is to replace the HD trajectories by "some other trajectories" that can sample the corresponding ensemble. In the case of the canonical ensemble, this is usually done by modifying the original HD such that they include the effect of the heat bath on the system under investigation. For this reason, these modifications are called *thermostats*. Commonly used thermostats are the Berendsen thermostat [13], the Nosé-Hoover thermostat [14], and the Langevin thermostat [8]. In this thesis, only the Langevin thermostat will be used for simulations. The reason for this choice is that for a huge class of potentials, the Langevin thermostat fulfils the ergodic hypothesis for sampling the canonical density [15–17]. For the often-used Nosé-Hoover thermostat it is possible to create cases where this is not the case [18, 19]. The same holds true for the Berendsen thermostat [20]. Furthermore, many proofs related to long-timescale algorithms assume Langevin type dynamics.

The Langevin thermostat models the effect of the heat bath by adding an additional stochastic term to the equations of motion. Therefore one no longer obtains deterministic trajectories as before but the evolution of the particle coordinates and velocities is now described by a stochastic process $\{(Q_t, V_t)\}$ where Q_t and V_t are $(n \times 3)$-dimensional random variables. This stochastic process must fulfil the *Langevin equations* [21]

$$dQ_t = V_t dt \tag{2.10}$$

$$dV_t = -M^{-1}\nabla E(Q_t)\, dt - \xi V_t\, dt + d\eta_t. \tag{2.11}$$

The effect of the heat bath is modelled by the stochastic process η_t that has zero mean, autocorrelation

$$\langle (\eta_t)_i (\eta_{t+\tau})_j \rangle = 2\frac{k_B T}{M_{i,i}} \xi \delta(\tau) \delta_{ij} \tag{2.12}$$

and no correlation with Q_s or V_s for $s \leq t$. The parameter $\xi \in \mathbb{R}$ is called the friction rate. Given some initial coordinates and velocities, one can solve the Langevin equations, i.e. use them to sample random trajectories. These trajectories can be used to calculate equilibrium observables with respect to the canonical density [22].

It is useful to observe that setting ξ to 0 would result in the deterministic Hamiltonian equations. Conversely, in the limit $\xi \to \infty$ one obtains the *overdamped Langevin equations* [23]

$$dQ_t = -\nabla E(Q_t)\, dt + d\nu_t. \tag{2.13}$$

The noise term ν_t is once again a stochastic process that has zero mean, autocorrelation

$$\langle (\nu_t)_i (\nu_{t+\tau})_j \rangle = 2k_B T \delta(\tau) \delta_{ij}, \tag{2.14}$$

and no correlation with Q_s for $s \leq t$. If the overdamped Langevin equation is used, only particle coordinates can be obtained but no velocities. While this is not sufficient to calculate ensemble

averages of the canonical ensemble, these trajectories may be used to calculate averages with respect to the canonical density in position space (see equation (2.8)).

2.5 Metastability and long-timescale simulations

As shown in the previous section, Langevin-based MD may be used to calculate ensemble averages for the canonical ensemble. Depending on the convergence speed of the time averages, very long trajectories may be necessary to obtain a good approximation.

Thermostatted dynamics have also been used for applications other than sampling the canonical density. As the thermostats are designed to model the effect of the heat bath, they have been used to investigate various dynamic properties of atomistic systems, like diffusion rates, reaction mechanisms or phase transitions. As shown in [24, 25], the dynamics that are obtained by solving the Langevin equations are in many cases similar to the dynamics produced by a full atomistic representation of the heat bath.

As in the HD case, it is usually not possible to calculate Langevin trajectories that are longer than a few microseconds, which may not be long enough to simulate the system of interest. The reasons for that may be manifold: low temperatures will slow the particle movements, due to the relation between temperature and kinetic energy, thus all chemical processes will take a much longer time. A high dimensionality of the phase space (i.e. many particles) is another problem. It raises the computational cost of each timestep in an MD simulation and may slow the convergence of the MD averages.

One particular problem is metastable behaviour of a particle system. From an observational point of view, this means that the dynamics of an atomistic system become stuck in some part of the phase space and will need a very long time to escape from it. A classical example for such behaviour is the simulation of a single one-dimensional particle in a double well potential as shown in figure 2.1. The time that the particle needs to escape from one of the potential wells has been analysed in great detail.

In his famous work, Kramers [26] investigated the rate with which the particle crosses the barrier both for the Langevin equations as well as the overdamped Langevin equations. At low temperatures, these rates can be used to accurately approximate the average time t_{escape} that the particle needs to escape from one of the wells [27]

$$t_{escape} \approx \frac{2\pi}{-E''(r)E''(s)} \exp\left(\frac{E(s) - E(r)}{k_B T}\right). \tag{2.15}$$

Here, $r \in \mathbb{R}$ is the coordinate of the local minimum from which the particle should escape whereas $s \in \mathbb{R}$ is the position of the barrier. One important observation is that the average escape time t_{escape} grows exponentially with respect to the inverse of the temperature. At low temperatures, it may no longer be possible to calculate Langevin trajectories that are long enough for the particle to cross the barrier.

Similar behaviour can be observed in a number of other, more complex simulations. In

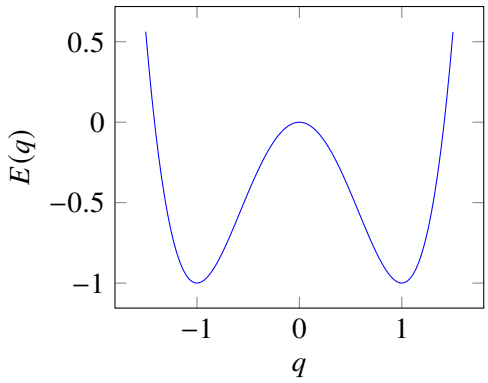


Figure 2.1: Double well potential.

the evaluation chapter 5, a number of more realistic problems are presented that also show a similar behaviour due to the presence of energy barriers. A more detailed explanation of the mathematical concept behind metastability as well as corresponding algorithms are presented in the next chapter.

CHAPTER 3

Coarse-grained long-timescale simulations

If a system exhibits metastability it may be difficult to calculate sufficiently long Langevin trajectories. However, metastability can make it possible to calculate coarse grained particle dynamics over larger timescales than would otherwise be the case. In this chapter, the necessary theory for such coarse grained dynamics as well as corresponding algorithms are presented.

The very basic idea that is behind most of these algorithms was developed before the advent of statistical mechanics. Historically, chemists have investigated chemical reactions in which some initial compounds (*reactants*) react to form some other (*product*) compounds. In the 19th century, Svante Arrhenius developed the Arrhenius equation [28]. This equation

$$k(T) = k(T_{ref}) \exp\left[\frac{E_a}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right]$$

models the connection between the reaction rate $k(T_{ref})$ at a given reference temperature T_{ref} and the reaction rate $k(T)$ at some other temperature. Here, R is the universal gas constant and E_a the *experimental activation energy* [29]. The Arrhenius equation is an empirical observation and values like the experimental activation energy have not been derived from the statistical mechanics ansatz that was presented before.

Still, it is tempting to use the concept of reactions and reaction rates to represent the dynamics of a particle system as a series of reactions that transfer the system from one metastable region to another.

3.1 Discretising the position space

If this idea is transferred to a particle system that moves according to the overdamped Langevin equations, this implies that the evolution of the particle system should be modelled as a sequence of jumps from one part of the position space to another. The common approach to realise this is to use a Markov Jump Process (MJP) to model the dynamics on a coarser scale. Such a MJP on

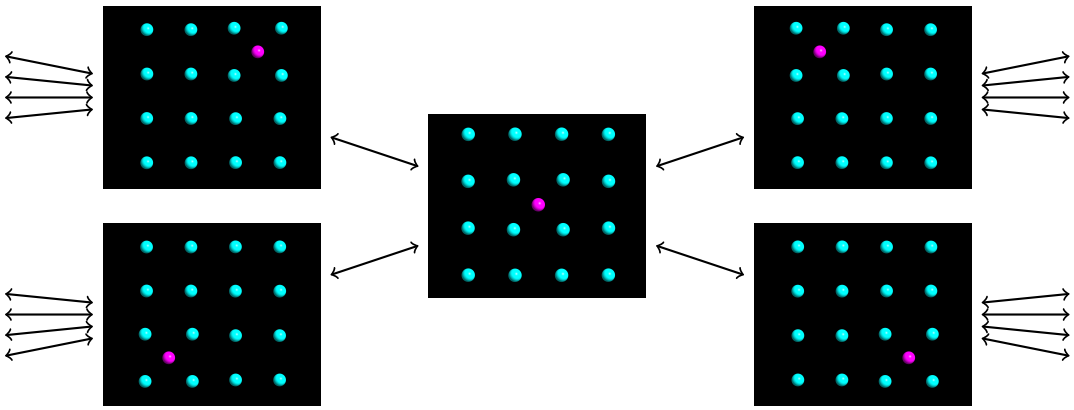


Figure 3.1: Representation of the evolution of an impurity atom as a MJP.

a (yet to be determined) finite state space $S = \{s_1, \ldots, s_m\}$ yields a graph-like representation of the position space of a particle system (see figure 3.1). Each state s_i of the MJP should correspond to some subset of the position space Θ_n which in turn should represent a metastable region. The evolution of the system is then modelled by a series of jumps from one state to the other. In the context of particle simulations, the rates that determine the jump frequency from one state s_i to some other state s_j are called the reaction or transition rates k_{ij}. Together with the values $k_{ii} = -\sum_{j \neq i} k_{ij}$ the transition rates form an $m \times m$ matrix that is called the rate matrix and which corresponds to the infinitesimal generator of the MJP.

These transition rates are central in governing the coarsened dynamics of the system. If a system is in some state s_i, the time that the system needs to escape from this state is an exponentially distributed random variable with parameter

$$K_i = \sum_{j \neq i} k_{ij}. \tag{3.1}$$

which is also called the total escape rate from state s_i. In section 3.2 a more detailed overview is given on how dynamical information can be extracted from a MJP.

One important question is whether simulating the dynamics by a MJP actually reproduces some properties of the underlying overdamped Langevin dynamics. Quite often the following argument is used: if a system has metastable regions, any system that enters such a metastable region will rapidly lose its memory about how it entered this region, leading to almost Markovian behaviour.

A proper mathematical analysis of this question was done in [23]. In this work, the key idea is to investigate the quasi stationary distribution (QSD) inside some metastable region $S \subset \Theta_n$ with respect to the overdamped Langevin process Q_t. QSDs are "distributions that are

invariant under time evolution when the process is conditioned to survive" [30] in the region S. A probability measure ν_S is a QSD on S if its support is S and if it fulfils [23]

$$\nu_S(A) = \frac{\int_S \mathbb{P}^q(Q_t \in A, t < \tau_S)\, \nu_S(dq)}{\int_S \mathbb{P}^q(t < \tau_S)\, \nu_S(dq)}, \quad \forall t > 0,\ \forall A \subset S.$$

Here, τ_S is the first exit time from S, i.e. $\tau_S = \inf\{t \geq 0, Q_t \notin S\}$ and $\mathbb{P}^q$ is "the probability measure under which $Q_0 = q$" [23]. If Q_0 in the overdamped Langevin equation (2.13) is distributed according to a QSD ν_S, the first exit time is exponentially distributed.

If Q_0 is distributed according to some other distribution μ_0 with support in S, it will converge to a unique QSD ν_S in the following sense:

$$\lim_{t\to\infty} \mathrm{Law}(Q_t | \tau_S > t) = \nu_S.$$

The crucial question is how fast the left side of the equation converges towards ν_S. It was shown in [23] that the convergence speed depends on the first two eigenvalues λ_1 and λ_2 of the eigenvalue problem

$$\begin{aligned} div(\nabla E + k_B T \nabla) u &= -\lambda u && \text{on } S \\ u &= 0 && \text{on } \partial S. \end{aligned}$$

Then the exit time is of order λ_1^{-1} and the time needed to converge to the QSD is of order $(\lambda_2 - \lambda_1)^{-1}$. Therefore the escape from the set S can be modelled as an exponentially distributed escape event if

$$\lambda_1^{-1} \gg (\lambda_2 - \lambda_1)^{-1}.$$

This analysis requires that the states of the MJP and the corresponding subsets of the position space are known in advance. In practice, it is important to know how one can determine the states and transition rates of such a MJP. Several theories that try to answer this question have been developed and some of them are presented now.

3.1.1 Transition state theory

Probably the most famous theory to determine the transition rates is the transition state theory (TST) that is usually traced back to the work of Eyring in 1935 [31]. In Eyring's theory, the position space Θ_n is split into three pairwise disjunct sets: the open reactant set R and the open product set P, separated by the dividing surface S such that $R \cup P \cup S = \Theta_n$. Then the TST transition rate from R to P is given by [32]

$$k_{R,P}^{TST} = \sqrt{\frac{k_B T}{2\pi}} \frac{\int_S \exp(-E(x)/(k_B T)) d\sigma(x)}{\int_R \exp(-E(x)/(k_B T)) dx} \tag{3.2}$$

where $d\sigma(x)$ is the surface element on S.

One important assumption of the TST is that every trajectory that leaves the reactant set R is reactive, i.e. as soon as it crosses the dividing surface it will stay in P until a local equilibrium is reached. In general this is not the case and a trajectory may repeatedly cross the dividing surface before settling in the product set. Therefore, the TST rate $k_{R,P}^{TST}$ always overestimates the actual transition rate [32].

TST itself does not define the decomposition of the position space into the sets R, P and S. The choice of the decomposition is however crucial, as a bad choice of the dividing surface may result in a transition rate that massively overestimates the actual transition rate [32, 33].

3.1.2 Harmonic transition state theory

The transition state theory is a powerful tool that provides the basic concepts needed for calculating transition rates. Unfortunately, calculating the transition rates as in equation (3.2) is usually not possible as it requires calculating high-dimensional integrals. Additionally, the question of how to define states is not answered by the classical TST.

To overcome this problem, a simplified version of TST, the harmonic transition state theory (HTST, also called Vineyard formula) [34] can be used. In the HTST, each state s_i of the MJP corresponds to a local minimiser x_i of the potential energy surface E. Two states s_i and s_j of the MJP are connected if a path between the corresponding local minimisers x_i and x_j exists that passes through one first order saddle point τ_{ij}. To calculate the transition rates, HTST

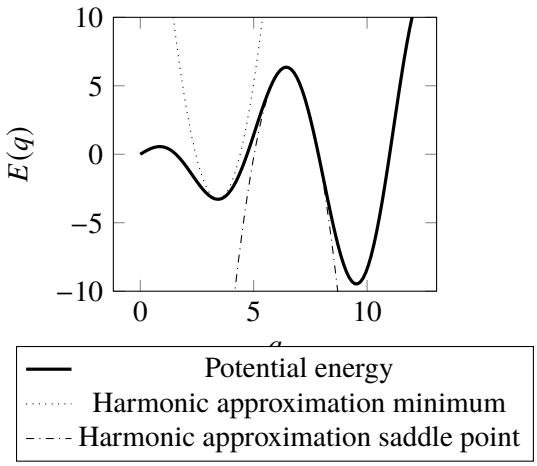


Figure 3.2: Harmonic approximations in one dimension.

assumes that the potential energy function E can be approximated by a quadratic function close to x_i and τ_{ij}, as shown in figure 3.2. For two states s_i and s_j the HTST transition rate is then

given by

$$k_{ij}^{HTST} = \nu_{ij} \exp(-\Delta E_{ij}/(k_B T)).$$

Here, $\Delta E_{ij} = E(\tau_{ij}) - E(x_i)$ is the *energy barrier*, i.e. the difference in energy between the initial energy minimum and the saddle point τ_{ij}. The factor ν_{ij} is called the *harmonic prefactor* and is given by

$$\nu_{ij} = \left(\prod_{k=1}^{3n} \mu_k(x_i)\right) / \left(\prod_{k=1}^{3n-1} \mu_k(\tau_{ij})\right). \tag{3.3}$$

Here, $\mu_k(x_i)$ denotes the k-th normal frequency of the potential energy function at x_i. Similarly, $\mu_k(\tau_{ij})$ is the k-th non-imaginary normal frequency at τ_{ij}.

Using the HTST it becomes much easier to build the corresponding MJP. As states correspond to local minimisers of the potential energy function, all such local minimisers, as well as first order saddle points that connect two minimisers must be found. As HTST was directly derived from TST, it inherits all its inaccuracies. Furthermore, the assumption that the potential energy function can be approximated by quadratic functions close to its saddle points and local minimisers introduces additional errors. HTST can be be unreliable at high temperatures and while it works well for some crystalline materials, this may not be the case for biomolecules with rugged energy landscapes [35, 36].

3.1.3 Other theories

Besides these TST based theories, a number of other ideas have been proposed to discretise the position space. As mentioned before, TST assumes that as soon as a particle system crosses the dividing surface between two states, it will not recross the surface but will stay in the other state for a long time. The *transition path theory (TPT)* [37] therefore looks at the distribution of reactive trajectories and uses it to calculate transition rates.

Another approach is used in *conformational dynamics (CD)* [38]. CD investigates metastability and transition rates by analysing the eigenvalues and eigenvectors of the so-called forward transfer operator. As this approach requires calculating the eigenvalues and eigenvectors of the forward transfer operator, it is limited to applications of low dimensionality, for example biomolecules that are essentially described by a low number of bond lengths or angles.

3.2 The Kinetic Monte Carlo algorithm

Given a Markov Jump Process that encodes the coarsened dynamics of a particle system, how can trajectories be extracted from it? One option is to use the *Kinetic Monte Carlo* (KMC) algorithm, that got its name from the work by Voter [39]. This algorithm is related to other similar algorithms like the *Gillespie algorithm* [40] or the *dynamic Monte Carlo* method.

The KMC algorithm (see flowchart in figure 3.3) can be used to sample trajectories from a given MJP. Here, a trajectory is a sequence $(\sigma_i, t_i)_i$ of states $\sigma_i \in S$ and corresponding transition

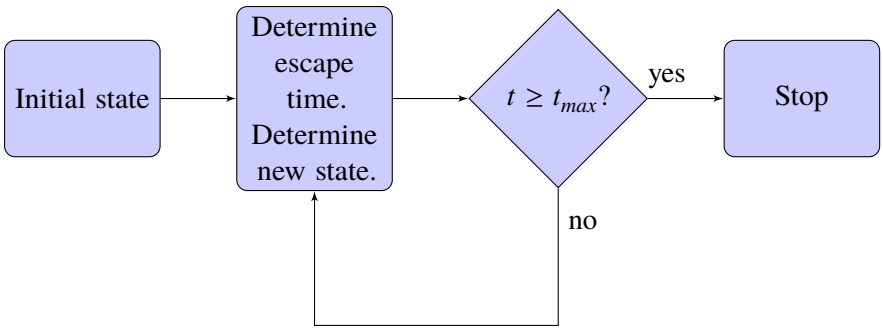


Figure 3.3: Flowchart of the KMC algorithm.

times t_i. The transition times determine the points in time when the system transitions from state σ_{i-1} to σ_i. The initial time t_0 is set to 0.

In a KMC simulation, such a trajectory is iteratively built, starting with some initial state σ_0 and time $t_0 = 0$. To determine the time that the system needs to escape from the state $\sigma_i = s_j$ as well as the state to which it escapes, the transition rates k_{jl} are used: for each state s_l that is connected to s_j, an exponentially distributed random number t_l^{escape} with parameter k_{jl} is drawn. By using the transition that corresponds to the shortest escape time ($l^* = \arg\min_l t_l^{escape}$) the new state and transition time are determined

$$\begin{aligned} \sigma_{i+1} &= s_{l^*} \\ t_{i+1} &= t_i + t_{l^*}^{escape} \end{aligned}$$

and the procedure is repeated until some final time t_{max} is reached. In most KMC implementations, the calculation of the new state and escape time is usually done in a computationally cheaper but mathematically equivalent way, see [39]. Instead of drawing exponentially distributed values for all connected states, only a single uniformly distributed random number is needed. The corresponding new state and transition time are determined afterwards by a binary search on the cumulative transition rates

$$C_{j,k} := \sum_{\substack{m=1 \\ m \neq j}}^{k} k_{jm}.$$

3.3 On-the-fly Kinetic Monte Carlo algorithms

One of the most restricting properties of the classical KMC algorithm is the fact that the full MJP and therefore all states and transitions must be known in advance. In most cases this is not the case and – depending on the size of the system under investigation – it may also not be possible to find all states and transitions.

For such problems *on-the-fly Kinetic Monte Carlo* algorithms have been developed[1]. Unlike the classical KMC algorithm, these methods initially know only one single state of the underlying MJP and will search for new states and transitions on the fly. This means that whenever the KMC algorithm evolves the system into a state that has not been visited before, all transitions that are possible from this state are determined. Afterwards the classical KMC procedure is used to advance the system to some other state, see figure 3.4.

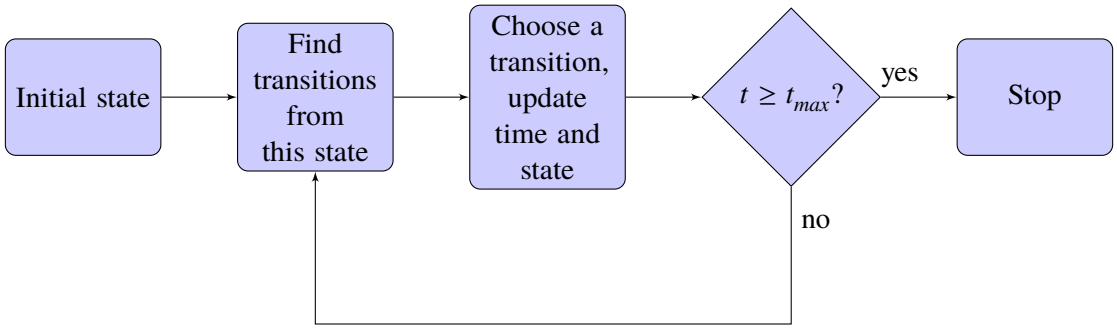


Figure 3.4: Flowchart of the on-the-fly KMC algorithm.

On-the-fly KMC methods must be able to find previously unknown states of the underlying MJP that are connected to some known state by a transition. In most cases, the harmonic transition state theory is used to define states and transition rates as it is computationally feasible even in high dimensions. Therefore, finding new transitions that are possible from some given state is essentially equal to finding nearby local minima and first order saddle points of the potential energy function.

Various approaches have been used for this task. In [41], high temperature Langevin trajectories were used to escape from the initial potential energy minimum followed by subsequent saddle searches. The ART nouveau method was used in [42] to find all first order saddle points around a given local minimum, whereas in [43] dimer searches were used for the same task.

3.4 Accelerated Dynamics Methods

The different KMC variants increase the accessible timescale by resorting to a discrete representation of the position space. Algorithms that belong to the class of *accelerated dynamic methods* (ADM) use a slightly different approach: in order to prevent a system from getting stuck in a metastable region, ADMs modify the (Langevin) dynamics in different ways such that it becomes easier for systems to escape from this region.

[1] On-the-fly KMC algorithms are known under a variety of different names like *adaptive KMC*, *self-evolving KMC* or *off-lattice KMC*.

These modifications change the dynamics of a system and would – if not treated properly – lead to wrong simulation results. Therefore, ADMs check whether the system leaves the initial metastable region and use some transition theory (usually TST or HTST) to calculate the time that the system would have needed to escape from the metastable state without the modification of the dynamics.

One of the first ADM methods was the *hyperdynamics* method [44]. It modifies the dynamics by adding additional *bias potentials* to the potential energy function E. Bias potentials are additional potential terms that are used to lower the height of the energy barriers as shown in figure 3.5. Once a system escapes from an energy minimum, HTST is used to renormalise the simulation time.

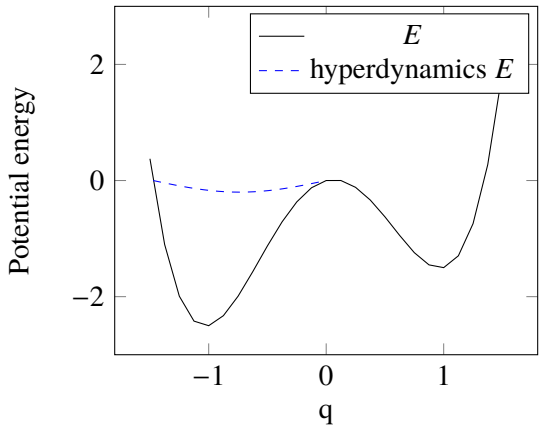


Figure 3.5: Modified potential energy in a hyperdynamics simulation.

Another simple way to improve the escape time from potential wells is to simply raise the simulation temperature. This idea, combined once again with a time rescaling based on HTST yields the *temperature-accelerated dynamics* (TAD) method [36]. Somewhat of an outlier is the *parallel replica dynamics* method [45]. Unlike hyperdynamics or TAD it does not increase the speed with which a system escapes from a potential minimum. Instead, it provides a way to parallelise this escape over a large amount of processor cores, thereby providing some kind of time-parallelisation for rare event simulations.

CHAPTER 4

Localised on-the-fly Kinetic Monte Carlo algorithms

On-the-fly KMC type algorithms have been used successfully in a number of different applications [46–48]. However, their high computational cost can prevent the simulation of other highly interesting phenomena. Due to the design of these algorithms (see figure 3.4), the majority of computational time is spent searching for transitions that are possible from a given state. Especially for larger or more complex systems, the time that is required to find these may become prohibitively large.

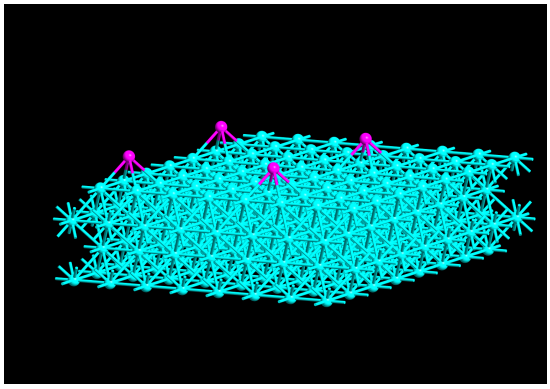


Figure 4.1: A number of ad-atoms (purple) on a crystal surface.

An example that demonstrates some of the problems of on-the-fly KMC algorithms is shown in figure 4.1: on a crystal surface a number of so-called ad-atoms (coloured purple) are placed – a situation that may occur in certain growth process simulations. As the time advances, these ad-atoms may move and form clusters or islands with other ad-atoms. The resulting surface structure is of interest as it may have a crucial effect on the properties of the final material.

Changing the number of ad-atoms has a profound impact on the resulting simulation: if all ad-atoms move according to some common transition mechanisms, the number of possible transitions grows linearly with the number of ad-atoms. Therefore, the total escape rate from the state also grows linearly with the number of ad-atoms, which causes the average time that the system stays in this state to decrease. Thus, the larger the number of ad-atoms, the more time is spent on transition searches while covering much less time per KMC step – a problem for efficient calculations.

While this example illustrates some of the problems of on-the-fly KMC algorithms, it may also offer a possible solution. It can be observed that – at least in the beginning of such a simulation – the movement of the ad-atoms are independent from each other. Only when two ad-atoms get close, will they start to have a non-negligible influence onto each other. This behaviour suggests that at least for these kinds of problems a localisation approach might be applicable. Localisation may bring down the cost of one single transition search search (by restricting it to the area of interest) while also lowering the number of necessary searches.

In section 4.1 the theoretical foundation for localising on-the-fly KMC algorithms is established. Afterwards, different localisation approaches are presented. In the remaining part of this chapter a localised on-the-fly KMC algorithm is presented that is capable of recycling known local transitions. It is is based on the k-ART algorithm and features a new comparison procedure for atomic neighbourhoods.

Please note that from now on it is no longer assumed that particles have three-dimensional coordinates. Instead, a more general approach is chosen and particles are simply assumed to be d-dimensional for some $d \in \mathbb{N}$.

4.1 Basic localisation theory

The common idea behind existing localised on-the-fly KMC algorithms [49–52] is that many particles can be ignored when looking for possible transitions. Only particles that initially lie in certain parts of an atomistic system, called *active regions*, are considered. Figure 4.2 shows an example: only parts of the system where the atoms do not form a perfect crystal structure are included in the active regions. The active regions and the corresponding particles inside them are defined as follows.

Definition 7. *Let $E : \mathbb{R}^{n\times d} \to \mathbb{R}$ be the potential energy function and $x \in \mathbb{R}^{n\times d}$ a local minimiser of E. Given some set $S \subseteq \mathbb{R}^d$, the set of* active indices *is defined by*

$$I(S, x) = \left\{ i \in \{1, \ldots, n\} : x_i \in S \right\}.$$

The set S is called the active region*. Similarly, the set of* fixed indices *is defined by*

$$F(S, x) = \{ i \in \{1, \ldots, n\} : i \notin I(S, x) \}.$$

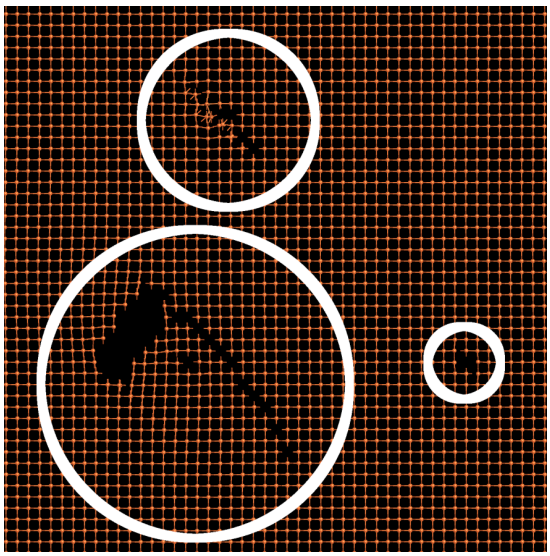


Figure 4.2: Active regions (marked by white circles) in a particle system.

Given some coordinates $y \in \mathbb{R}^{n\times d}$*, the notation*

$$y_{F(S,x)} \in \mathbb{R}^{|F(S,x)|\times d}$$

is used to denote the coordinates of the particles which indices are in the fixed index set.

Now assume that some particle system is in an energy minimum, i.e. its coordinates $x \in \mathbb{R}^{n\times d}$ are a local minimiser of the potential energy function E. Any (HTST-based) non-localised on-the-fly KMC algorithm would now try to find all possible transitions from this state by determining all nearby local minimisers of E that are connected to x by a first order saddle point. In a localised setting, instead of finding local minimisers of the unconstrained minimisation problem

$$\min_{p\in\mathbb{R}^{n\times d}} E(p) \tag{4.1}$$

the idea is to find local minimisers of the constrained problem

$$\min_{p\in\mathbb{R}^{n\times d}} E(p) \text{ s.t. } p_{F(S,x)} = x_{F(S,x)}. \tag{4.2}$$

This corresponds to holding all particles fixed that do not initially lie inside the active region $S \subseteq \mathbb{R}^d$.

The crucial question is of course: if $p \neq x$ is another local minimiser of the minimisation problem in (4.1), does the constrained problem (4.2) have a local minimiser $\tilde{p}$ that is in some

sense close to p? The answer to this question determines whether a transition search can be localised. If equation (4.2) has a local minimiser that is close enough to p then this local minimiser can be used as a good approximation to it. An initial result can be derived from the following theorem.

Theorem 1. *Let $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ be a function with $(x, y) \in \mathbb{R}^m \times \mathbb{R}^n$ a local minimiser such that $\nabla f(x, y) = 0$ and the Hessian matrix Hf is positive definite at (x, y). It is also assumed that f is twice continuously differentiable in an open neighbourhood around (x, y).*

Then an open neighbourhood $N \subseteq \mathbb{R}^m$ around 0 as well as a unique and continuously differentiable function $g : N \to \mathbb{R}^n$ exist, such that $g(\epsilon)$ is a strict local minimiser of

$$\min_{z \in \mathbb{R}^n} \underbrace{f(x + \epsilon, z)}_{=:h_\epsilon(z)}$$

for all $\epsilon \in N$.

Proof. The proof is a simplified version of the proof for [53, Theorem 2.1].

Due to the assumptions on f and (x, y), the implicit function theorem can be applied immediately to $\nabla_y f(x + \epsilon, y) = \nabla h_\epsilon(y)$. It shows that an open neighbourhood $\tilde{N}$ and a unique and continuously differentiable function $g : \tilde{N} \to \mathbb{R}^n$ exist, such that $g(\epsilon)$ is a critical point of $h_\epsilon(z)$ for all $\epsilon \in \tilde{N}$. Thus it remains to be shown that some open neighbourhood $N \subseteq \tilde{N}$ around 0 exist, such that the Hessian matrix $Hh_\epsilon(g(\epsilon))$ is positive definite for all $\epsilon \in N$. By construction, $Hh_\epsilon(z)$ is given by

$$Hh_\epsilon(z) = [Hf(x + \epsilon, z)]_{m+1:m+n, m+1:m+n} .$$

As $Hh_\epsilon(z)$ is a principal submatrix of $Hf(x + \epsilon, z)$, its smallest eigenvalue cannot be smaller than the smallest eigenvalue of $Hf(x + \epsilon, z)$ [54]. Due to the assumptionthat $Hf(x, y)$ is positive definite and f twice continuously differentiable close to (x, y), $Hh_\epsilon(g(\epsilon))$ is positive definite for some open neighbourhood $N \subseteq \tilde{N}$. □

This theorem can be applied to the previous localisation idea. Once again let $S \subseteq \mathbb{R}^d$ be an active region and $x \in \mathbb{R}^{n \times d}$ and $p \in \mathbb{R}^{n \times d}$ two strict local minimisers of the potential energy function E. W.l.o.g. it is assumed here that the fixed index set $F(S, x)$ is given by $F(S, x) = \{1, \ldots, m\}$ for some m which can always be achieved by simply reordering the coordinates. By splitting the coordinates into active and fixed coordinates, the constrained optimisation problem

$$\min_{y \in \mathbb{R}^{n \times d}} E(y) \text{ s.t. } y_{F(S,x)} = p_{F(S,x)} + \epsilon \tag{4.3}$$

can be rewritten as

$$\min_{z \in \mathbb{R}^{|I(S,x)| \times d}} E(p_{F(S,x)} + \epsilon, z). \tag{4.4}$$

In the context of local transition searches, ϵ is usually set to $\epsilon = x_{F(S,x)} - p_{F(S,x)}$, i.e. the difference in the fixed coordinates between the initial local minimiser x and p. Due to theorem 1, if $\epsilon \in \mathbb{R}^{|F(S,x)|\times d}$ is sufficiently small (i.e. $\epsilon \in N$), the minimisation problem in (4.4) has $g(\epsilon)$ as a local minimiser where g is the unique function in theorem 1. As g is continuously differentiable, it is Lipschitz continuous on N and thus

$$\|g(\epsilon) - g(0)\| = \left\|g(\epsilon) - p_{I(S,x)}\right\| \leq K \|\epsilon\|$$

with K being the Lipschitz constant with respect to some chosen norm. Hence, under the same conditions $\tilde{p} \coloneqq (p_{F(S,x)} + \epsilon, g(\epsilon))$ is a local minimiser of the minimisation problem in (4.3) and

$$\|\tilde{p} - p\|_F = \left\| \begin{pmatrix} p_{F(S,x)} + \epsilon - p_{F(S,x)} \\ g(\epsilon) - g(0) \end{pmatrix} \right\|_F \leq K \|\epsilon\|_F + \|\epsilon\|_F = (1 + K) \|\epsilon\|_F .$$

Thus theorem 1 basically offers a very basic error estimate as well as a first stability result for the localisation approach.

It is worth pointing out that the assumptions in theorem 1 are usually fulfilled in most practical simulations. Many classical potential energy functions are twice continuously differentiable almost everywhere [1] if short-range interactions are truncated using a sufficiently smooth tapering approach.

While theorem 1 shows that the minima searches can be localised under certain circumstances, the same does not seem to be true for saddle searches. One simple counter example would be the function

$$E(x) = x^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} x$$

that has a first order saddle point at $x = 0$. Fixing x_3 to some arbitrary value $a \in \mathbb{R}$ results in $G_a(x_1, x_2) = E(x_1, x_2, a) = x_1^2 + x_2^2 + a^2$ which has no first order saddle point anymore. Still, in most practical cases it still seems to be possible to localise many saddle searches that way as will be shown in section 5.2.

4.2 Localised on-the-fly Kinetic Monte Carlo algorithms

As mentioned, the idea to localise the transition-search phase in on-the-fly KMC algorithms is not new and has been used previously. One of the first algorithms to use localisation is the *self-evolving atomistic kinetic Monte Carlo* algorithm [49]. In each KMC step, it determines the active regions (called active volumes here) where transitions are possible, using different heuristics. Afterwards, the possible transitions inside the active volumes are determined.

[1] They are usually not differentiable for coordinates $x \in \mathbb{R}^{n\times d}$ that correspond to a particle system where two particles have the same coordinates.

Another localisation ansatz was used in the *kinetic activation relaxation technique* (k-ART) [50, 55] as well as in the (unnamed) algorithm in [51]. Instead of using active volumes of various sizes, these algorithms decompose atomistic systems into different *atomic neighbourhoods*. The atomic neighbourhood of an atom is the collection of all the atoms that are located inside a certain radius r_{cut} around the given *central atom* as shown in figure 4.3. To describe an atomic neighbourhood, the following definition is used:

Definition 8. *The atomic neighbourhood N (or for short: neighbourhood) of some particle in a particle system is defined by the tuple* $N = (q, e) = (q_1, \ldots, q_n, e_1, \ldots, e_n)$ *where* $q_i \in \mathbb{R}^d$ *are the d-dimensional coordinates of the n particles relative to the central atom. The* e_i *define the elements of those particles.*

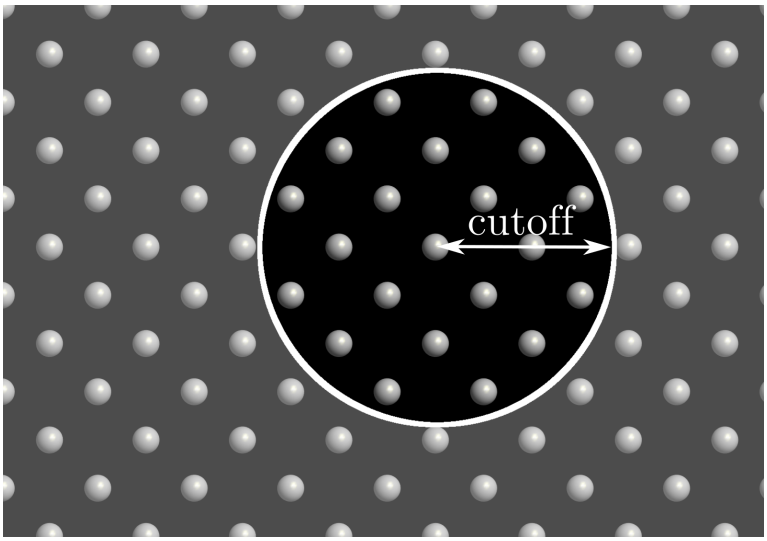


Figure 4.3: An atomic neighbourhood of an atom in a larger system.

Thus whenever all possible transitions in an n-particle system are needed, k-ART decomposes it into the n corresponding neighbourhoods $N_1, \ldots, N_n$, using some given cutoff radius r_{cut}. For each neighbourhood, the algorithm checks whether this neighbourhood has been encountered before and if information about the local transitions is available, using a database of atomic neighbourhoods (more about this in the next section). For all new neighbourhoods, local transitions that may occur inside them are calculated and stored in the database. This local transition information is finally used to calculate the corresponding global transitions. A summary of k-ART can be found in the form of a flowchart in figure 4.4.

An important feature of k-ART is that the local transition information is stored in a database. Therefore, local transitions need not be recalculated if the corresponding neighbourhood has been encountered before, thus saving computational time. This addresses another problem of

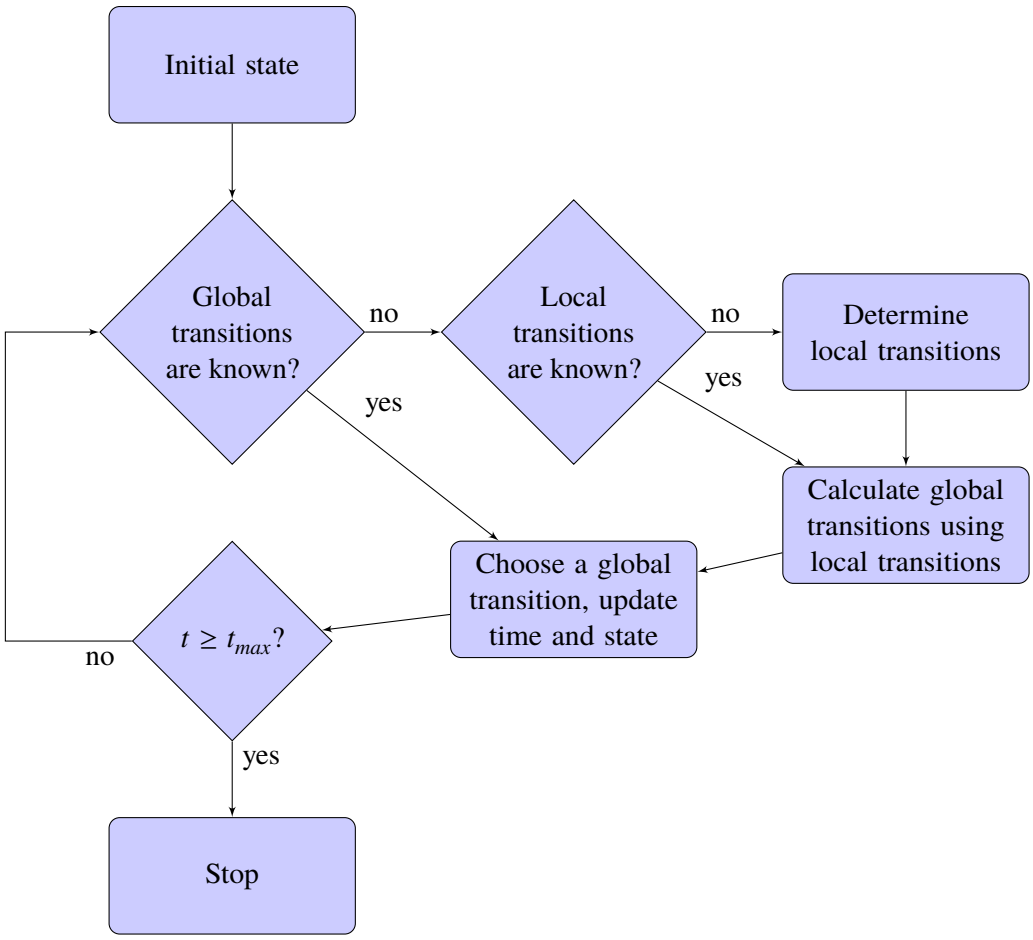


Figure 4.4: Flowchart of a k-ART simulation.

classical on-the-fly KMC algorithms: their inability to reuse known transition mechanisms. In a lot of simulations, similar transitions occur in different regions of the system under investigation, as in the example in the beginning of this chapter (figure 4.1). If the ad-atoms are separated from each other, they will evolve independently using the same reaction mechanisms until they get close enough that they start to interact. Standard on-the-fly KMC algorithms have to find the possible transitions for each ad-atom individually and are not capable of reusing the already calculated transition mechanism. On-the-fly KMC algorithms that are able to reuse known transition information are called *self-learning on-the-fly KMC* algorithms. Examples are the already mentioned algorithm in [51] and k-ART. On a simpler level, some on-the-fly KMC algorithms are able to *recycle* known transitions. Unlike self-learning KMC algorithms, these algorithms try to reuse the transition mechanisms from previous transition searches without checking the current structure of the system. This means that they simply apply known

transition mechanisms to the current state and check if the resulting transitions are actually valid. Examples for this recycling procedure can once again be found in [56].

Even in non-KMC algorithms, localisation has been used. A local variant of the hyperdynamics method was presented in [57] in order to overcome issues when applying hyperdynamics to larger systems. The motivation for this localisation ansatz was that the speed advantage of hyperdynamics over a classical MD simulation vanishes with increasing system sizes [57]. In [52] a localised saddle search approach was presented that can be used to find local transitions in some active region.

4.3 Comparison functions for atomic neighbourhoods

One of the key ideas of k-ART is to reuse local transition information that is already known. This requires some kind of database which stores atomic neighbourhoods and the corresponding local transitions that may occur inside them. Before discussing the intricacies of efficient database structures a much more basic question must be answered: *When are two neighbourhoods similar?* This question is of fundamental importance as its answer determines how previously encountered neighbourhoods can be detected.

A naive approach would be to see two neighbourhoods $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ as similar if their Cartesian coordinates and elements match exactly, i.e.

$$q = \tilde{q}, \quad e = \tilde{e}.$$

The usefulness of this approach is rather limited for a number of reasons. To begin with, exact matches are rare in realistic simulations. The neighbourhoods of two atoms in a particle system may be almost equal but have slightly different coordinates due to numerical inaccuracies as well as small distortions that are caused by the structure of the remainder of the system.

Another problem is that the atoms in an atomic neighbourhood have no fixed order or orientation. Rotating a neighbourhood or permuting the particle indices does not change its structure. If no external forces are present, rotating[2] and permuting an atomic neighbourhood in this way will result in a neighbourhood whose local transitions that are equal to the rotated and permuted local transitions of the original neighbourhood. A similar behaviour occurs also in molecules. If once again no external forces are present, the potential energy of a molecule is invariant under index permutations, orthogonal transformations and translations. Other properties, such as forces, can be mapped from one molecule to some transformed version of this molecule using these three different transformation types. This behaviour can be derived from QM where the potential energy of molecules is also invariant under these transformations [58].

As local transition searches can be computationally expensive, it would be advantageous to recycle as many local transitions as possible. Being able to recognise two atomic neighbourhoods as similar, even if one is a slightly distorted, permuted and rotated instance of the other is

[2] In general one can use not only rotations but orthogonal transformations.

therefore beneficial. If these two neighbourhoods are seen as similar, this would also allow k-ART to reuse the local transitions of one of the neighbourhoods for the other if the corresponding transformations are known. The question whether these invariances are really necessary is investigated later in subsection 5.3.5.

Thus, in order to obtain an efficient k-ART implementation, some kind of comparison function $C(N, \tilde{N})$ is needed, that determines whether two neighbourhoods N and $\tilde{N}$ are similar or not. If N and $\tilde{N}$ are similar, $C(N, \tilde{N})$ should be 1, otherwise 0. To be of use, such a comparison function should have certain properties. It should be invariant under orthogonal transformations and permutations, i.e. for two neighbourhoods $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ it should fulfil

$$C(N, \tilde{N}) = C((q, e), (\tilde{q}, \tilde{e})) = C((q, e), (R\tilde{q}_{\pi_1}, \ldots, R\tilde{q}_{\pi_n}, \tilde{e}_{\pi_1}, \ldots, \tilde{e}_{\pi_n}))$$

for all orthogonal transformations R and permutations π. It should also satisfy some standard conditions like symmetry

$$C(N, \tilde{N}) = C(\tilde{N}, N)$$

and for all neighbourhoods, the following equation should be true

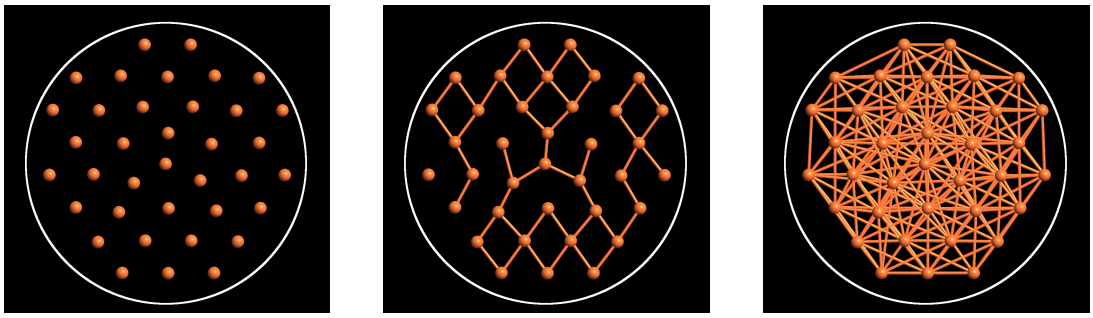
$$C(N, N) = 1.$$

While the comparison function is needed for finding similar neighbourhoods it is also necessary to know how two similar neighbourhoods are related to each other, i.e. how can one neighbourhood be mapped onto the other. This is important as soon as a local transition that belongs to one neighbourhood should be applied to another similar (but transformed) neighbourhood.

In the original k-ART papers [50, 55], a graph based approach was used for the comparison. First, each neighbourhood is transformed into an undirected graph, where each node represents a single particle. Two particles/nodes are then connected by an edge, if the distance between these particles is smaller than some given cutoff radius r_{graph} (see figure 4.5).

This cutoff radius has to be chosen carefully (usually close to the bond length) as wrong values might lead to useless graphs. For $r_{graph} = 0$ the resulting graph will be the null graph, while cutoffs that are larger than the diameter of the neighbourhood will produce full graphs. Two neighbourhoods are then seen as similar if their corresponding graphs are equal up to permutation of the indices. This comparison is usually done using the *nauty* [59, 60] software package. Nauty is capable of calculating a label for each graph such that two graphs are similar if their labels are equal (canonical label).

While k-ART with the nauty-based comparison function has been shown to work well on a number of problems, it is possible to create cases where the comparison of two neighbourhoods fails [61, 62]. As shown in figure 4.6, two neighbourhoods that are structurally very different, may be seen as equal as their corresponding graphs are similar. This behaviour, known as false positives, may assign wrong local transitions to a neighbourhood. The k-ART algorithm tries to avoid this problem by modifying the cutoff radii on the fly if false positives are detected [63].



(a) bond length times 0 (b) bond length times 1.1. (c) bond length times 2.5

Figure 4.5: Adjacency graphs for different cutoff radii.

Another problem is that the k-ART comparison function does not offer any information about the transformations that map two neighbourhoods onto each other [62]. Thus, in k-ART an additional heuristic mapping scheme is used if two neighbourhoods are similar [64].

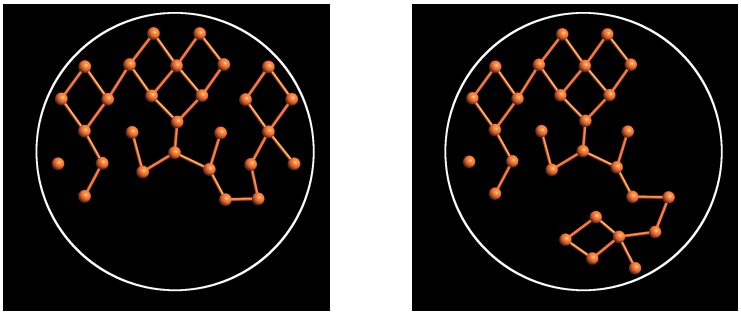


Figure 4.6: Two structurally different atomic neighbourhoods with the same adjacency graph.

Alternative comparison functions have been proposed. In [51], neighbourhoods are compared by looking at the number of particles in so-called shells (spherical regions around the central particle) in both neighbourhoods. If they are (almost) equal, the two neighbourhoods are designated as similar. Like the k-ART comparison function, this approach may produce false positives and is not capable of calculating the optimal transformations.

A different approach was chosen in [65] where different distance functions $d(N, \tilde{N})$ were used to build a comparison function of the form

$$C(N, \tilde{N}) = \begin{cases} 1, & d(N, \tilde{N}) \leq \epsilon \\ 0, & d(N, \tilde{N}) > \epsilon. \end{cases} \tag{4.5}$$

The problem with this approach is to find distance functions $d(N, \tilde{N})$ that are computable, do not cause false negatives/positives and that offer information about the optimal transformations. In [65] two different distance functions where used. One was called the *overlap of Gaussian type orbitals* and – while being invariant under permutations and rotations – did not offer any information about the involved transformations. The other was based on an invariant version of the root mean square deviation but could not be calculated exactly. In the next section, the latter function is used once again and an alternative way to calculate it exactly is shown.

4.4 The invariant RMSD comparison function

Equation (4.5) is now used to build a comparison function for k-ART. Thus, a suitable distance function $d(N, \tilde{N})$ is needed. Especially in machine learning applications a large variety of such functions that are invariant under orthogonal transformations and permutations have been used. Examples are the SOAP kernel [66], the bispectrum [67] or a combination of Coulomb matrices and Gaussian kernels [68]. Related functions were also introduced in [65] and [69]. Unfortunately, these approaches do not offer any information on the optimal transformations and therefore cannot be used in k-ART without having to resort to some heuristic to map local transitions from one neighbourhood to the other.

Therefore, a different kind of distance function is used here. It is based on the well-known *root mean square deviation of atomic positions (RMSD)*, which for two neighbourhoods $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ is given by

$$d^{RMSD}(N, \tilde{N}) = \begin{cases} \infty & e \neq \tilde{e} \\ \sqrt{\sum_{i=1}^{n} \left\| q_i - \tilde{q}_i \right\|_2^2} & e = \tilde{e}. \end{cases} \tag{4.6}$$

Here, both N and $\tilde{N}$ are supposed to be atomic neighbourhoods with n d-dimensional particles each. The RMSD in equation (4.6) is not invariant under orthogonal transformations and permutations. However, it is easy to create invariant versions of it. First, define $\mathbb{O}_d$ to be the set of all possible $d \times d$ orthogonal matrices. Similarly, S_n is the set of all possible permutations of size n. For simplicity, the notation e_π is used to to denote a permutation of e by π, i.e.

$$(e_\pi)_i = e_{\pi_i} \quad i = 1 \ldots, n.$$

Then an RMSD that is invariant under orthogonal transformations can be created by minimising over all orthogonal transformations:

$$d^{orth}(N, \tilde{N}) = \begin{cases} \infty & e \neq \tilde{e} \\ \min_{R \in \mathbb{O}_d} \sqrt{\sum_{i=1}^{n} \left\| q_i - R\tilde{q}_i \right\|_2^2} & e = \tilde{e}. \end{cases} \tag{4.7}$$

Similarly, a permutation-invariant version of the RMSD can be build

$$d^{perm}(N, \tilde{N}) = \min_{\pi \in S_n} \begin{cases} \infty & e \neq \tilde{e}_\pi \\ \sqrt{\sum_{i=1}^{n} \left\| q_i - \tilde{q}_{\pi_i} \right\|_2^2} & e = \tilde{e}_\pi, \end{cases} \tag{4.8}$$

as well as a version that is invariant under both orthogonal transformations and permutations

$$d^{full}(N, \tilde{N}) = \min_{\pi \in S_n} \begin{cases} \infty & e \neq \tilde{e}_\pi \\ \min_{R \in \mathbb{O}_d} \sqrt{\sum_{i=1}^{n} \left\| q_i - R\tilde{q}_{\pi_i} \right\|_2^2} & e = \tilde{e}_\pi. \end{cases} \tag{4.9}$$

These RMSD variations can not only be used to compare atomic neighbourhoods but also molecules and – by ignoring the elements – point clouds. When the elements are ignored, all these RMSD based distance functions are pseudometrics on $\mathbb{R}^{n \times d}$.

None of these invariant RMSDs are new and all of them have seen extensive use before. Calculating the orthogonally invariant RMSD for point sets in (4.7) is equal to the *orthogonal procrustes problem* and can be solved using the Kabsch algorithm [70] in linear time with respect to the number of points. If the particles are three-dimensional, a quaternion based approach can also be used to find the optimal orthogonal matrix [71]. Calculating the permutationally invariant RMSD can be reduced to the *assignment problem* in combinatorics and is therefore solvable in cubic time [72] with respect to the number of points.

As mentioned before, the fully (permutation + orthogonal transformation) invariant RMSD in equation (4.9) has been presented before in [65] but also in [69, 73] as well as in numerous computer vision articles, where it is used in the point set registration problem [74–76].

For a fixed dimension d, calculating d^{full} in polynomial time with respect to n is in general not possible [73]. Nevertheless, a number of algorithms to calculate it as efficiently as possible have been proposed which can roughly be divided into two classes. The first class contains algorithms for particle systems [65, 73] as well as for point sets [74, 77] that calculate only approximations to d^{full}. As d^{full} is not calculated exactly, false positives/negatives may occur when such an algorithm is used to compare atomic neighbourhoods in k-ART. A second class of methods has emerged in recent years and consists of algorithms that calculate d^{full} exactly [69, 75, 76]. Depending on the problem under investigation, these algorithms may have excessive computational cost.

However, k-ART does not require the calculation of $d^{full}(N, \tilde{N})$ for general neighbourhoods. Due to equation (4.5) it is sufficient to decide whether

$$d^{full}(N, \tilde{N}) \leq \epsilon \tag{4.10}$$

holds true or not. If it does, the optimal transformations should be calculated.

A number of exact algorithms were tested to see if they are sufficiently fast for this task (see the evaluation in section 5.3). As none of them was, I developed a new algorithm that

can determine whether (4.10) holds true or not in polynomial time (with respect to n) if ϵ is sufficiently small. If the inequality in (4.10) holds, it is also capable of calculating the orthogonal transformation and permutation that minimises (4.9).

4.5 Calculating the invariant RMSD

In this section it is shown, how to determine whether $d^{full}(N, \tilde{N}) \leq \epsilon$ holds true for two atomic neighbourhoods $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$. To begin with, some trivial cases are excluded.

First of, it is assumed that both neighbourhoods contain n particles and that a permutation $\pi \in S_n$ exists, such that $e = \tilde{e}_\pi$. If this is not the case[3], $d^{full}(N, \tilde{N}) = \infty$ by its definition. Otherwise, $d^{full}(N, \tilde{N}) \leq \epsilon$ is equivalent to

$$f(N, \tilde{N}) := \min_{\pi \in S_n : e = \tilde{e}_\pi} \min_{R \in \mathbb{O}_d} \sum_{i=1}^{n} \left\| q_i - R\tilde{q}_{\pi_i} \right\|_2^2 \leq \epsilon^2$$

and $d^{full}(N, \tilde{N}) = \sqrt{f(N, \tilde{N})}$. For simplicity, $g(N, \tilde{N}, R, \pi)$ is defined to be

$$g(N, \tilde{N}, R, \pi) = \sum_{i=1}^{n} \left\| q_i - R\tilde{q}_{\pi_i} \right\|_2^2.$$

Thus $f(N, \tilde{N})$ can also be written as

$$f(N, \tilde{N}) = \min_{\pi \in S_n : e = \tilde{e}_\pi} \min_{R \in \mathbb{O}_d} g(N, \tilde{N}, R, \pi). \tag{4.11}$$

Furthermore, it is assumed here that at least one of the neighbourhoods has full rank. A neighbourhood $N = (q, e)$ of n d-dimensional particles is said to have full rank if q contains d linearly independent coordinates. Otherwise, $d^{full}(N, \tilde{N})$ can be calculated on a lower dimension which is proven in theorem 2. Before that, an auxiliary lemma is needed:

Lemma 1. *Let $q = (q_1, \ldots, q_n)$ $(q_i \in \mathbb{R}^d)$ and $\tilde{q} = (\tilde{q}_1, \ldots, \tilde{q}_n)$ $(\tilde{q}_i \in \mathbb{R}^d)$ be two sets of n d-dimensional points. Then the orthogonal transformation R^* that minimises*

$$\min_{R \in \mathbb{O}_d} \sum_{i=1}^{n} \left\| q_i - R\tilde{q}_{\pi_i} \right\|_2^2 \tag{4.12}$$

for an arbitrary permutation $\pi \in S_n$ is given by

$$R^* = UV^T$$

[3] Which can easily be verified in $O(n \log n)$ by sorting e and $\tilde{e}$.

where $U\Sigma V^T$ is the singular value decomposition of

$$A = \sum_{i=1}^{n} q_i \tilde{q}_{\pi_i}^T.$$

Furthermore, R^* is a global minimiser.

Proof. A proof can be found in the work of Kabsch [70] who is also the namesake of the Kabsch algorithm that is used to solve the minimisation problem in equation (4.12). □

Theorem 2. *Let $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ be two atomic neighbourhoods with n particles of dimension d each. Denote by r and $\tilde{r}$ the rank of the matrices*

$$Q = \begin{bmatrix} | & & | \\ q_1 & \dots & q_n \\ | & & | \end{bmatrix}$$

and

$$\tilde{Q} = \begin{bmatrix} | & & | \\ \tilde{q}_1 & \dots & \tilde{q}_n \\ | & & | \end{bmatrix}.$$

Assume that $\max(r, \tilde{r}) < d$ and w.l.o.g. $r \geq \tilde{r}$ (otherwise switch N and $\tilde{N}$). Let $Q = U\Sigma V^T$ and $\tilde{Q} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$ be the singular value decompositions of Q and $\tilde{Q}$ and define $p = (p_1, \dots, p_n)$ $(p_i \in \mathbb{R}^r)$ and $\tilde{p} = (\tilde{p}_1, \dots, \tilde{p}_n)$ $(\tilde{p}_i \in \mathbb{R}^r)$ by

$$p_i = (U^T q_i)_{1:r}, \quad \tilde{p}_i = (\tilde{U}^T \tilde{q}_i)_{1:r}.$$

If $\tilde{R}$ and $\tilde{\pi}$ are a global minimum of

$$\min_{\pi \in S_n : e = \tilde{e}_\pi} \min_{R \in \mathbb{O}_r} g((p, e), (\tilde{p}, \tilde{e}), R, \pi)$$

then $\tilde{\pi}$ and[4]

$$R^* = U \begin{bmatrix} \tilde{R} & 0_{r \times d-r} \\ 0_{d-r \times r} & I_{d-r} \end{bmatrix} \tilde{U}^T$$

form a global minimum of

$$\min_{\pi \in S_n : e = \tilde{e}_\pi} \min_{R \in \mathbb{O}_d} g(N, \tilde{N}, R, \pi)$$

and

$$g(N, \tilde{N}, R^*, \tilde{\pi}) = g((p, e), (\tilde{p}, \tilde{e}), \tilde{R}, \tilde{\pi}).$$

[4] $0_{r \times d-r}$ denotes the $r \times d - r$ zero matrix. I_{d-r} is the $(d - r) \times (d - r)$ unit matrix.

Proof. Let $\pi \in S_n$ be a permutation such that $e = \tilde{e}_\pi$ and let $\hat{U}(\pi)\hat{\Sigma}(\pi)\hat{V}(\pi)^T$ be the SVD of

$$\hat{A}(\pi) = \sum_{i=1}^{n} q_i \tilde{q}_{\pi_i}^T$$

and similarly $\dot{A}(\pi) = \dot{U}(\pi)\dot{\Sigma}(\pi)\dot{V}(\pi)^T$ for

$$\dot{A}(\pi) = \sum_{i=1}^{n} p_i \tilde{p}_{\pi_i}^T.$$

Using lemma 1 it can be established that $\hat{R}(\pi) = \hat{U}(\pi)\hat{V}(\pi)^T$ is a global minimiser of

$$\min_{R \in \mathbb{O}_d} g(N, \tilde{N}, R, \pi) \tag{4.13}$$

and $\dot{R}(\pi) = \dot{U}(\pi)\dot{V}(\pi)^T$ of

$$\min_{R \in \mathbb{O}_r} g((p, e), (\tilde{p}, \tilde{e}), R, \pi). \tag{4.14}$$

It is now shown that $\hat{R}(\pi)$ fulfils

$$\hat{R}(\pi) = U \begin{bmatrix} \dot{R}(\pi) & 0_{r \times d-r} \\ 0_{d-r \times r} & I_{d-r} \end{bmatrix} \tilde{U}^T. \tag{4.15}$$

First, one can observe that $U^T q_i$ and $\tilde{U}^T \tilde{q}_i$ fulfil

$$U^T q_i = \begin{pmatrix} p_i \\ 0_{d-r} \end{pmatrix}, \quad \tilde{U}^T \tilde{q}_i = \begin{pmatrix} \tilde{p}_i \\ 0_{d-r} \end{pmatrix}.$$

This is due to

$$U^T q_i = U^T Q_{*,i} = (\Sigma V^T)_{*,i}.$$

As Q has rank $r < d$, the last $d - r$ rows of Σ (and therefore the last $d - r$ entries of $U^T q_i$) are

zero and $(U^T q_i)_{1:r}$ is equal to p_i by definition. Therefore $\hat{A}$ can be written as

$$\begin{aligned}
\hat{A}(\pi) &= \sum_{i=1}^{n} q_i \tilde{q}_{\pi_i}^T \\
&= \sum_{i=1}^{n} U \begin{pmatrix} p_i \\ 0_{d-r} \end{pmatrix} \begin{pmatrix} \tilde{p}_{\pi_i} \\ 0_{d-r} \end{pmatrix}^T \tilde{U}^T \\
&= U \left[\sum_{i=1}^{n} \begin{pmatrix} p_i \\ 0_{d-r} \end{pmatrix} \begin{pmatrix} \tilde{p}_{\pi_i} \\ 0_{d-r} \end{pmatrix}^T \right] \tilde{U}^T \\
&= U \begin{bmatrix} \sum_{i=1}^{n} p_i \tilde{p}_{\pi_i}^T & 0_{r\times d-r} \\ 0_{d-r\times r} & 0_{d-r\times d-r} \end{bmatrix} \tilde{U}^T \\
&= U \begin{bmatrix} \dot{A}(\pi) & 0_{r\times d-r} \\ 0_{d-r\times r} & 0_{d-r\times d-r} \end{bmatrix} \tilde{U}^T.
\end{aligned}$$

Then it can easily be verified that

$$\begin{aligned}
\hat{U}(\pi) = U \begin{bmatrix} \dot{U}(\pi) & 0_{r\times d-r} \\ 0_{d-r\times r} & I_{d-r} \end{bmatrix}, \hat{\Sigma}(\pi) = \begin{bmatrix} \dot{\Sigma}(\pi) & 0_{r\times d-r} \\ 0_{d-r\times r} & 0_{d-r\times d-r} \end{bmatrix}, \\
\hat{V}(\pi) = \tilde{U} \begin{bmatrix} \dot{V}(\pi) & 0_{r\times d-r} \\ 0_{d-r\times r} & I_{d-r} \end{bmatrix}
\end{aligned}$$

is a valid SVD of $\hat{A}(\pi)$. Therefore the optimal orthogonal transformation $\hat{R}(\pi)$ is

$$\begin{aligned}
\hat{R}(\pi) = \hat{U}(\pi)\hat{V}(\pi)^T &= U \begin{bmatrix} \dot{U}(\pi) & 0_{r\times d-r} \\ 0_{d-r\times r} & I_{d-r} \end{bmatrix} \begin{bmatrix} \dot{V}(\pi)^T & 0_{r\times d-r} \\ 0_{d-r\times r} & I_{d-r} \end{bmatrix} \tilde{U}^T \\
&= U \begin{bmatrix} \dot{U}(\pi)\dot{V}(\pi)^T & 0_{r\times d-r} \\ 0_{d-r\times r} & I_{d-r} \end{bmatrix} \tilde{U}^T = U \begin{bmatrix} \dot{R}(\pi) & 0_{r\times d-r} \\ 0_{d-r\times r} & I_{d-r} \end{bmatrix} \tilde{U}^T.
\end{aligned}$$

Thus for any $\pi \in S_n$ that fulfils $e = \tilde{e}_\pi$ it has been established that if $\dot{R}(\pi)$ minimises equation (4.14), $\hat{R}(\pi)$ as in equation (4.15) minimises equation (4.13). It can also be shown that

$g(N, \tilde{N}, \hat{R}(\pi), \pi) = g((p, e), (\tilde{p}, \tilde{e}), \dot{R}(\pi), \pi)$:

$$\begin{aligned}
g(N, \tilde{N}, \hat{R}(\pi), \pi) &= \sum_{i=1}^{n} \left\| q_i - \hat{R}(\pi)\tilde{q}_{\pi_i} \right\|_2^2 \\
&= \sum_{i=1}^{n} \left\| U^T q_i - U^T \hat{R}(\pi)\tilde{U}\tilde{U}^T \tilde{q}_{\pi_i} \right\|_2^2 \\
&= \sum_{i=1}^{n} \left\| \begin{pmatrix} p_i \\ 0_{d-r} \end{pmatrix} - U^T \hat{R}(\pi)\tilde{U} \begin{pmatrix} \tilde{p}_{\pi_i} \\ 0 \end{pmatrix} \right\|_2^2 \\
&= \sum_{i=1}^{n} \left\| \begin{pmatrix} p_i \\ 0_{d-r} \end{pmatrix} - \begin{bmatrix} \dot{R}(\pi) & 0_{r\times d-r} \\ 0_{d-r\times r} & I_{d-r} \end{bmatrix} \begin{pmatrix} \tilde{p}_{\pi_i} \\ 0 \end{pmatrix} \right\|_2^2 \\
&= \sum_{i=1}^{n} \left\| p_i - \dot{R}(\pi)\tilde{p}_{\pi_i} \right\|_2^2 \\
&= g((p, e), (\tilde{p}, \tilde{e}), \dot{R}, \pi).
\end{aligned}$$

Due to this equality it does not matter whether the optimal permutation is calculated for $g(N, \tilde{N}, \hat{R}(\pi), \pi)$ or $g((p, e), (\tilde{p}, \tilde{e}), \dot{R}(\pi), \pi)$ and therefore $\tilde{\pi}$ minimises both terms. □

4.5.1 The case $\epsilon = 0$

Now that the prerequisites have been sorted out, it is time to start with the case $\epsilon = 0$, i.e. two neighbourhoods N and $\tilde{N}$ are seen as similar if

$$d^{full}(N, \tilde{N}) = 0$$

or equivalently

$$f(N, \tilde{N}) = 0.$$

This case is much simpler than the $\epsilon > 0$ case but shows the basic ideas that are also needed for $\epsilon > 0$.

Due to the construction of $f(N, \tilde{N})$ in equation (4.11), it can be calculated by an exhaustive search over all $O(n!)$-many permutations of length n. For large values of n this is of course not possible. If at least one of the neighbourhoods has no overlapping atoms (atoms with the same coordinates), the following theorem shows, however, that it is possible to check for $f(N, \tilde{N}) = 0$ by testing only $O(n^d)$-many permutations:

Theorem 3. *Let $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ be two neighbourhoods of n d-dimensional particles each such that $\tilde{q}$ has full rank d. Also assume that $q_k \neq q_l$ and $\tilde{q}_k \neq \tilde{q}_l$ for all $k \neq l$. Then let $j_1, \ldots, j_d$ be d indices such that $\tilde{q}_{j_1}, \ldots, \tilde{q}_{j_d}$ are linearly independent.*

If and only if $f(N, \tilde{N}) = 0$*, there exist* d *indices* $i_1, \ldots, i_d$ *such that* $g(N, \tilde{N}, \bar{R}, \bar{\pi}) = 0$ *for*

$$\bar{R} = \underset{R \in \mathbb{O}_d}{\arg\min} \sum_{k=1}^{d} \left\| q_{i_k} - R\tilde{q}_{j_k} \right\|_2^2 \tag{4.16}$$

and

$$\bar{\pi} = \underset{\pi \in S_n : e = \tilde{e}_\pi}{\arg\min} \sum_{k=1}^{n} \left\| q_k - \bar{R}\tilde{q}_{\pi_k} \right\|_2^2 . \tag{4.17}$$

Proof. Both directions of the *if and only if* part of the theorem are proven now. If $f(N, \tilde{N}) = 0$, there exist by definition transformations R^* and π^* such that $g(N, \tilde{N}, R^*, \pi^*) = 0$. This also implies that $q_i = R^* \tilde{q}_{\pi_i^*}$ for all i.

By choosing $i_1, \ldots, i_d$ such that $\pi^*_{i_k} = j_k$ $(k = 1, \ldots, d)$, it can be verified that R^* minimises equation (4.16) to zero. R^* is also the unique solution to this equation as

$$\sum_{k=1}^{d} \left\| q_{i_k} - R\tilde{q}_{j_k} \right\|_2^2 = 0$$

requires $q_{i_k} = R\tilde{q}_{j_k}$ for $k = 1, \ldots, d$. As the $\tilde{q}_{j_k}$ were assumed to be linearly independent, there exists only one such orthogonal transformation. Once again, π^* minimises (4.17) to zero and as it was assumed that no two particles inside an neighbourhood may have the same position, it is also a unique solution.

The other direction is trivial. As $g(N, \tilde{N}, \bar{R}, \bar{\pi}) = 0$, $\bar{R}$ and $\bar{\pi}$ can be used in the definition of f to show that $f(N, \tilde{N}) = 0$. □

Using this theorem, an algorithm can be derived that checks whether $f(N, \tilde{N}) = 0$ or not. This is done by choosing d indices $j_1, \ldots, j_d$ such that $\tilde{q}_{j_1}, \ldots, \tilde{q}_{j_d}$ are linearly independent. In the next step, all $\binom{n}{d}$-many tuples $i_1, \ldots, i_d$ are tested and the corresponding transformations $\bar{R}$ and $\bar{\pi}$ are calculated. Due to theorem 3, if $f(N, \tilde{N}) = 0$, one of these $i_1, \ldots, i_d$ will yield the globally optimal solution. When all things are put together, the algorithm 1 is obtained.

4.5.2 The case $\epsilon > 0$

In most applications, it is useful to allow small deviations in the particle positions when comparing atomic neighbourhoods. These deviations may stem from the limited numerical accuracy in computer simulations or simply from the fact, that some neighbourhoods are slightly affected by remote defects or impurities in a material. Thus it is beneficial to develop an algorithm that can run the check $f(N, \tilde{N}) \leq \epsilon^2$ even for *small* $\epsilon > 0$. As mentioned before, it is not possible to calculate this for arbitrarily large values ϵ, as this problem is NP-hard in general.

In almost all classical atomistic simulations one can assume that two particles will never get any closer than some distance μ, due to the strong repulsive forces between nearby particles.

Algorithm 1 Algorithm that checks whether $f(N, \tilde{N}) = 0$.

Require: Two neighbourhoods $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ that contain n d-dimensional particles.
Require: $\tilde{q}$ must have full rank.
Require: $q_k \neq q_l$ and $\tilde{q}_k \neq \tilde{q}_l \; \forall k \neq l$.
Require: $\exists \pi \in S_n$ such that $e = \tilde{e}_\pi$.
1: Choose d indices $j_1, \ldots, j_d$ such that $\tilde{q}_{j_1}, \ldots, \tilde{q}_{j_d}$ are linearly independent.
2: **for all** $i_1, \ldots, i_d : [(e_{i_k} = \tilde{e}_{j_k} \forall k = 1, \ldots, d) \wedge (\min_{R \in \mathbb{O}_d} \sum_{k=1}^{d} \left\| q_{i_k} - R\tilde{q}_{j_k} \right\|_2^2 = 0)]$ **do**
3: $\quad \bar{R} = \arg\min_{R \in \mathbb{O}_d} \sum_{k=1}^{d} \left\| q_{i_k} - R\tilde{q}_{j_k} \right\|_2^2$
4: $\quad \bar{\pi} = \arg\min_{\pi \in S_n : e = \tilde{e}_\pi} \sum_{k=1}^{n} \left\| q_k - \bar{R}\tilde{q}_{\pi_k} \right\|_2^2$
5: $\quad$ **if** $g(N, \tilde{N}, \bar{R}, \bar{\pi}) = 0$ **then**
6: $\quad\quad$ **return** Equal, $\bar{R}$, $\bar{\pi}$
7: $\quad$ **end if**
8: **end for**
9: **return** Not equal

Using such a lower bound for the pairwise particle distances, algorithm 1 can be modified such that it can deal with values $\epsilon \leq \mu/(2\sqrt{1+4d})$. The big difference to the $\epsilon = 0$ case is, that it is no longer possible to exactly calculate the optimal orthogonal matrix R^* using just d particles. Thus, a modified version of the algorithm 1 is used and forms algorithm 2.

There are a few major differences to the $\epsilon = 0$ algorithm. First of, the algorithm no longer chooses indices $j_1, \ldots, j_d$ such that $\tilde{q}_{j_1}, \ldots, \tilde{q}_{j_n}$ are just linearly independent. Instead, the $\tilde{q}_{j_1}, \ldots, \tilde{q}_{j_n}$ should be the "maximally linearly independent points" which means that they should maximise the determinant of the matrix that they form. Second, even if $f(N, \tilde{N}) \leq \epsilon^2$, the orthogonal matrix R that is calculated using d points (line 4) is in general not equal to the globally optimal orthogonal matrix for some indices $i_1, \ldots, i_d$. Still, if $f(N, \tilde{N}) \leq \epsilon^2$, one of the permutations $\bar{\pi}$ that is calculated in line 8 using these orthogonal matrices will be equal to the globally optimal permutation. Algorithm 2 can determine whether $f(N, \tilde{N}) \leq \epsilon^2$, assuming that the assumptions that were made in the beginning hold true. In order to verify these assumptions, algorithm 3 can be used. Now it remains to prove that the algorithm 2 does indeed calculate whether $f(N, \tilde{N}) \leq \epsilon^2$ and – if this is the case – that the returned transformations R^* and π^* are indeed optimal. The first step is to show the effects of the choice of the $j_1, \ldots, j_d$.

Algorithm 2 Algorithm that checks whether $f(N, \tilde{N}) \leq \epsilon^2$.

Require: Two neighbourhoods $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ that contain n d-dimensional particles.
Require: $\tilde{q}$ must have full rank.
Require: $\left\|\tilde{q}_k - \tilde{q}_l\right\|_2 \geq \mu \; \forall k \neq l$
Require: $q_k \neq q_l \; \forall k \neq l$.
Require: $\exists \pi \in S_n$ such that $e = \tilde{e}_\pi$.
Require: $\epsilon \leq \frac{\mu}{2\sqrt{1+4d}}$.
$g^* = \infty$, $R^* = I$, $\pi^* = id$
Choose d indices $j_1, \ldots, j_d = \arg\min_{j_1,\ldots,j_d} |\det\left[\tilde{q}_{j_1}, \ldots, \tilde{q}_{j_d}\right]|$.
for all $i_1, \ldots, i_d$ such that $(e_{i_k} = \tilde{e}_{j_k} \forall k = 1, \ldots, d)$ **do**
    $R = \arg\min_{R \in \mathbb{O}_n} \sum_{k=1}^{d} \left\|q_{i_k} - R\tilde{q}_{j_k}\right\|_2^2$
    if $\sum_{k=1}^{d} \left\|q_{i_k} - R\tilde{q}_{j_k}\right\|_2^2 > \epsilon^2$ **then**
        continue
    end if
    Set $\bar{\pi} \in S_n$ such that $\bar{\pi}_{i_k} = j_k$ for all $k = 1, \ldots, d$ and $\bar{\pi}_k = \arg\min_{l:\, e_k = \tilde{e}_l} \left\|q_k - R\tilde{q}_l\right\|_2^2$ for all $k \notin \{i_1, \ldots, i_d\}$.
    $\bar{R} = \arg\min_{R \in \mathbb{O}_n} \sum_{k=1}^{n} \left\|q_k - R\tilde{q}_{\bar{\pi}_k}\right\|_2^2$
    if $g(N, \tilde{N}, \bar{R}, \bar{\pi}) < g^*$ **then**
        $g^* = g(N, \tilde{N}, \bar{R}, \bar{\pi})$
        $R^* = \bar{R}$
        $\pi^* = \bar{\pi}$
    end if
end for
if $g^* \leq \epsilon^2$ **then**
    return Equal, g^*, R^*, π^*
else
    return Not equal
end if

Algorithm 3 Algorithm $comp^{full}(N, \tilde{N}, \epsilon)$ that checks whether $d(N, \tilde{N}) \leq \epsilon$.

Require: Two neighbourhoods $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ of size n and $\tilde{n}$.
1: $e^{sorted} = sort(e)$, $\tilde{e}^{sorted} = sort(\tilde{e})$.
2: **if** $e^{sorted} \neq \tilde{e}^{sorted}$ **then**
3: **return** $d^{full}(N, \tilde{N}) = \infty$
4: **end if**
5: $r = rank(q)$, $\tilde{r} = rank(\tilde{q})$
6: **if** $r > \tilde{r}$ **then**
7: **return** $comp^{full}(\tilde{N}, N, \epsilon)$
8: **end if**
9: **if** $\tilde{r} < d$ **then**
10: Reduce dimension of N and $\tilde{N}$.
11: **end if**
12: Calculate the largest μ that fulfils $\left\|q_i - q_j\right\|_2 \geq \mu \, \forall i \neq j$
13: Calculate the largest $\tilde{\mu}$ that fulfils $\left\|\tilde{q}_i - \tilde{q}_j\right\|_2 \geq \tilde{\mu} \, \forall i \neq j$
14: **if** $\mu > \tilde{\mu}$ and $r = \tilde{r}$ **then**
15: **return** $comp^{full}(\tilde{N}, N, \epsilon)$
16: **end if**
17: **if** $\epsilon > \frac{\tilde{\mu}}{2\sqrt{1+4d}}$ **then**
18: Fail
19: **end if**
20: **return** Result of algorithm 2

Theorem 4. *Let $\tilde{q} = (\tilde{q}_1, \ldots, \tilde{q}_n)$ ($\tilde{q}_i \in \mathbb{R}^d$) be n d-dimensional points and*

$$\tilde{Q} = \begin{bmatrix} | & & | \\ \tilde{q}_1 & \cdots & \tilde{q}_n \\ | & & | \end{bmatrix}$$

the corresponding point matrix. Assume that $\tilde{Q}$ has full rank and define d indices $j_1, \ldots, j_d$ as follows

$$j_1, \ldots, j_d = \underset{j_1, \ldots, j_d}{\arg\max} \left| \det\left(\begin{bmatrix} | & & | \\ \tilde{q}_{j_1} & \cdots & \tilde{q}_{j_d} \\ | & & | \end{bmatrix} \right) \right|.$$

Let $\tilde{S}$ be the corresponding matrix

$$\tilde{S} = \begin{bmatrix} | & & | \\ \tilde{q}_{j_1} & \cdots & \tilde{q}_{j_d} \\ | & & | \end{bmatrix}.$$

Then $\tilde{S}$ is invertible and the following inequality holds

$$\left\|\tilde{S}^{-1}\tilde{q}_i\right\|_\infty \leq 1, \quad \forall i = 1, \ldots, n. \tag{4.18}$$

This also implies that each $\tilde{q}_i$ can be written as

$$\tilde{q}_i = \tilde{S}\tilde{v}_i = \sum_{k=1}^{d} \tilde{q}_{j_k}(\tilde{v}_i)_k$$

where $\tilde{v}_i = \tilde{S}^{-1}\tilde{q}_i$ and $\left\|v_i\right\|_\infty \leq 1$.

Proof. As $\tilde{Q}$ was assumed to have full rank, the $d \times d$ submatrix $\tilde{S}$ also has full rank and is thereby invertible. Using Cramer's rule, the k-th component of $\tilde{S}^{-1}\tilde{q}_i$ is given by

$$(\tilde{S}^{-1}\tilde{q}_i)_k = \frac{\det \tilde{S}^{(i,k)}}{\det \tilde{S}}.$$

Here, $\tilde{S}^{(i,k)}$ is the matrix $\tilde{S}$ where the k-th column of $\tilde{S}$ has been replaced by $\tilde{q}_i$. As $\tilde{S}$ is the submatrix of $\tilde{Q}$ that has the largest determinant in absolute terms, the inequality

$$|\det \tilde{S}^{(i,k)}| \leq |\det \tilde{S}|$$

holds for all i and k, leading to

$$\left|(\tilde{S}^{-1}\tilde{q}_i)_k\right| = \frac{|\det \tilde{S}^{(i,k)}|}{|\det \tilde{S}|} \leq 1 \Rightarrow \left\|\tilde{S}^{-1}\tilde{q}_i\right\|_\infty \leq 1.$$

Please note that an equivalent theorem has already been stated in [78], though without a proof. □

Using this theorem, each point $\tilde{q}_i$ can be written as $\tilde{q}_i = \tilde{S}\tilde{v}_i$ where $\tilde{v}_i \in \mathbb{R}^d$ fulfils $\left\|\tilde{v}_i\right\|_\infty \leq 1$. This property is now used to bound the error that occurs when the algorithm tries to calculate the optimal orthogonal matrices.

Theorem 5. *Let $q = (q_1, \ldots, q_n)$ ($q_i \in \mathbb{R}^d$) and $\tilde{q} = (\tilde{q}_1, \ldots, \tilde{q}_n)$ ($\tilde{q}_i \in \mathbb{R}^d$) be two sets of n d-dimensional points. Assume that the $\tilde{q}_i$ fulfil the assumptions in theorem 4 and let $j_1, \ldots, j_d$ be the optimal indices in this theorem. Also let $R \in \mathbb{O}$ and $\tilde{R} \in \mathbb{O}$ be two orthogonal matrices, $i_1, \ldots, i_d$ d pairwise different indices and $\pi \in S_n$ a permutation that fulfils $\pi_{i_k} = j_k$ for all $k = 1, \ldots, d$. Define δ and $\tilde{\delta}$ by*

$$\delta := \sqrt{\sum_{k=1}^{d} \left\| q_{i_k} - R\tilde{q}_{\pi_{i_k}} \right\|_2^2}$$

and

$$\tilde{\delta} := \sqrt{\sum_{k=1}^{d} \left\| q_{i_k} - \tilde{R}\tilde{q}_{\pi_{i_k}} \right\|_2^2}.$$

Then the following inequality holds for all $l = 1, \ldots, n$:

$$\left\|(\tilde{R} - R)\tilde{q}_l\right\|_2 \leq \sqrt{d}(\delta + \tilde{\delta}).$$

Proof. Using $\sum_{i=1}^{d} |a_i| \leq \sqrt{d}\sqrt{\sum_{i=1}^{d} a_i^2}$ one obtains

$$\begin{aligned}
\sum_{k=1}^{d} \left\|(\tilde{R} - R)\tilde{q}_{j_k}\right\|_2 &= \sum_{k=1}^{d} \left\|(\tilde{R} - R)\tilde{q}_{j_k} + q_{i_k} - q_{i_k}\right\|_2 \\
&\leq \sum_{k=1}^{d} \left\|q_{i_k} - \tilde{R}\tilde{q}_{j_k}\right\|_2 + \left\|q_{i_k} - R\tilde{q}_{j_k}\right\|_2 \\
&\leq \sqrt{d \sum_{k=1}^{d} \left\|q_{i_k} - \tilde{R}\tilde{q}_{j_k}\right\|_2^2} + \sqrt{d \sum_{k=1}^{d} \left\|q_{i_k} - R\tilde{q}_{j_k}\right\|_2^2} \\
&= \sqrt{d}(\delta + \tilde{\delta}).
\end{aligned}$$

Due to theorem 4, every $\tilde{q}_l$ can be written as $\tilde{q}_l = \tilde{S}\tilde{v}_l = \sum_{k=1}^{d} \tilde{v}_{lk}\tilde{q}_{j_k}$ where $\left\|v_l\right\|_\infty \leq 1$. Thus for every $l = 1, \ldots n$ one can write

$$\begin{aligned}
\left\|(\tilde{R} - R)\tilde{q}_l\right\|_2 &= \left\|(\tilde{R} - R)\tilde{S}\tilde{v}_l\right\|_2 \\
&= \left\|(\tilde{R} - R)\sum_{k=1}^{d} \tilde{v}_{lk}\tilde{q}_{j_k}\right\|_2 \\
&\leq \sum_{k=1}^{d} \left\|(\tilde{R} - R)\tilde{v}_{lk}\tilde{q}_{j_k}\right\|_2 \\
&\leq \sum_{k=1}^{d} |\tilde{v}_{lk}| \left\|(\tilde{R} - R)\tilde{q}_{j_k}\right\|_2 \\
&\leq \sum_{k=1}^{d} \left\|(\tilde{R} - R)\tilde{q}_{j_k}\right\|_2 \\
&\leq \sqrt{d}(\delta + \tilde{\delta}).
\end{aligned}$$

□

Finally, it is necessary to show that if $f(N, \tilde{N}) \leq \epsilon^2$, algorithm 2 will at some point calculate

the globally optimal permutation in line 8. This in turn will cause the algorithm to calculate the globally optimal orthogonal matrix in the subsequent line.

Theorem 6. *Let $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ be two neighbourhoods (that contain n d-dimensional particles each) where the $\tilde{q}_1, \ldots, \tilde{q}_n$ fulfil*

$$\left\|\tilde{q}_i - \tilde{q}_j\right\|_2 \geq \mu, \; \forall i \neq j.$$

Furthermore, assume that the $\tilde{q}_i$ have full rank (as in theorem 4) and that $j_1, \ldots, j_d$ are the corresponding indices in this theorem that maximise the determinant. Then if $f(N, \tilde{N}) \leq \epsilon^2$ for

$$\epsilon < \frac{\mu}{2\sqrt{1+4d}},$$

there exist indices $i_1, \ldots, i_d$ such that the globally optimal permutation $\pi^* \in S_n$ is equal to $\bar{\pi} \in S_n$ that is defined by

$$\begin{aligned} \bar{\pi}_{i_k} &= j_k & & k = 1, \ldots, d \\ \bar{\pi}_k &= \underset{l:e_k=\tilde{e}_l}{\arg\min} \left\|q_k - R\tilde{q}_l\right\|_2 & & k \notin \{i_1, \ldots, i_d\} \end{aligned}$$

where $R \in \mathbb{O}_d$ is given by

$$R = \min_{R \in \mathbb{O}_d} \sum_{k=1}^{d} \left\|q_{i_k} - R\tilde{q}_{j_k}\right\|_2^2.$$

Proof. Let $i_1, \ldots, i_d$ be defined implicitly by $\pi^*_{i_k} = j_k$ $(k = 1, \ldots, d)$ and let $R^* \in \mathbb{O}_d$ be the globally optimal orthogonal matrix that (together with π^*) minimises $f(N, \tilde{N})$. In that case $\bar{\pi}_{i_k} = j_k = \pi^a st_{i_k}$ for all $k = 1, \ldots, d$. Thus it must now be proven that π^*_k is equal to $\bar{\pi}_k = \arg\min_{l:e_k=\tilde{e}_l} \left\|q_k - R\tilde{q}_l\right\|_2$ for all other indices $k \notin \{i_1, \ldots, i_d\}$. Similar to theorem 5, δ and δ^* are defined as

$$\delta = \sqrt{\sum_{k=1}^{d} \left\|q_{i_k} - R\tilde{q}_{\pi_{i_k}}\right\|_2^2}$$

and

$$\delta^* := \sqrt{\sum_{k=1}^{d} \left\|q_{i_k} - R^*\tilde{q}_{\pi_{i_k}}\right\|_2^2}.$$

Then by definition $\delta \leq \delta^* \leq \epsilon$ and, due to theorem 5, the following inequality holds for all $k = 1, \ldots, n$:

$$\left\|(R - R^*)\tilde{q}_k\right\| \leq \sqrt{d}(\delta + \delta^*) \leq 2\sqrt{d}\delta^*.$$

For every $k \notin \{i_1, \ldots, i_d\}$ one also obtains

$$\left\| q_k - R^* \tilde{q}_{\pi_k^*} \right\|_2 \leq \sqrt{\epsilon^2 - \delta^{*2}}$$

due to

$$\begin{aligned}
\epsilon^2 \geq f(N, \tilde{N}) &= \sum_{l=1}^{n} \left\| q_l - R^* \tilde{q}_{\pi_l^*} \right\|_2^2 \\
&= \sum_{l=1}^{d} \left\| q_{i_l} - R^* \tilde{q}_{j_l} \right\|_2^2 + \sum_{\substack{l=1 \\ l \notin \{i_1, \ldots, i_d\}}}^{n} \left\| q_l - R^* \tilde{q}_{\pi_l^*} \right\|_2^2 \\
&= \delta^{*2} + \sum_{\substack{l=1 \\ l \notin \{i_1, \ldots, i_d\}}}^{n} \left\| q_l - R^* \tilde{q}_{\pi_l^*} \right\|_2^2 \\
&\geq \delta^{*2} + \left\| q_k - R^* \tilde{q}_{\pi_k^*} \right\|_2^2 .
\end{aligned}$$

Using R instead of R^* yields for every $k \notin \{i_1, \ldots, i_d\}$

$$\begin{aligned}
\left\| q_k - R\tilde{q}_{\pi_k^*} \right\|_2 &= \left\| q_k - R\tilde{q}_{\pi_k^*} + R^*\tilde{q}_{\pi_k^*} - R^*\tilde{q}_{\pi_k^*} \right\|_2 \\
&\leq \left\| q_k - R^*\tilde{q}_{\pi_k^*} \right\|_2 + \left\| (R - R^*)\tilde{q}_{\pi_k^*} \right\|_2 \\
&\leq \sqrt{\epsilon^2 - \delta^{*2}} + 2\sqrt{d}\delta^* \\
&\leq \sqrt{1 + 4d}\epsilon.
\end{aligned} \tag{4.19}$$

The last inequality can be derived by minimising $\sqrt{\epsilon^2 - \delta^{*2}} + 2\sqrt{d}\delta^*$ with respect to $\delta^* \in [0, \epsilon]$. The inequality in (4.19) shows that $R\tilde{q}_{\pi_k^*}$ lies in a sphere of radius $\sqrt{1 + 4d}\epsilon$ around q_k. As it was assumed that $\epsilon < \frac{\mu}{2\sqrt{1+4d}}$ (and thus $\mu > 2\sqrt{1 + 4d}\epsilon$), no other $R\tilde{q}_l$ lies inside this sphere around q_k. Thus $\bar{\pi}_k$, defined by

$$\bar{\pi}_k = \underset{l : e_k = \tilde{e}_l}{\arg\min} \left\| q_k - R\tilde{q}_l \right\|_2$$

is equal to π_k^* for all $k \notin \{i_1, \ldots, i_d\}$. □

4.5.3 Computational complexity

It is of course of interest to investigate the computational complexity of algorithm 2 that checks whether $f(N, \tilde{N}) \leq \epsilon^2$. The different components of the algorithms are analysed now. As

before, it is assumed that both neighbourhoods N and $\tilde{N}$ have n d-dimensional particles.

Calculating optimal orthogonal transformations

An operation that is needed several times in the algorithm is the calculation of the orthogonal transformation that minimises

$$\min_R \sum_{i=1}^{n} \left\| q_i - R\tilde{q}_i \right\|_2^2$$

for some given points q_i and $\tilde{q}_i$ of dimension d. Using a SVD-based approach as in [70] (see also lemma 1), this requires three steps. First, the $d \times d$ matrix

$$A = \sum_{i=1}^{n} q_i \tilde{q}_i^T$$

is computed. Afterwards, the SVD of A is calculated, i.e. $A = U\Sigma V^T$. The optimal orthogonal matrix is then given by $R = UV^T$. Calculating the matrix A requires $O(nd^2)$ operations. The subsequent calculation of the SVD can be realised in $O(d^3)$ and the calculation of $R = UV^T$ takes once again $O(d^3)$ operations.

Putting all things together, the computational complexity of the full minimisation is of order $O(nd^2 + d^3)$. Calculating the corresponding value of the objective function takes time $O(nd^2)$ and thus does not change the overall complexity.

Finding the optimal indices (line 2)

The first part of the algorithm involves the calculation of the indices $j_1, \ldots, j_d$ that maximise the absolute value of the determinant of $\tilde{S} = [\tilde{q}_{j_1}, \ldots, \tilde{q}_{j_d}]$. Algorithms that generate approximate solutions for this problem exist [78] but in order to obtain the exact globally optimal solution, a brute-force search over all combinations has been used in this work.

This requires an exhaustive search over all $\binom{n}{d}$ ($O(n^d)$) index combinations of length d. For each index combination, the corresponding determinant of a $d \times d$ matrix must be calculated, which requires $O(d^3)$ operations when using a LU decomposition[5]. Thus, a brute force search to determine the optimal indices has computational complexity $O(n^d d^3)$.

[5] It is possible to calculate the determinant with complexity $O(d^{2.373})$ using a fast matrix multiplication algorithm [79] and a corresponding matrix factorisation [80]. While these algorithms are asymptotically faster than the classical $O(d^3)$ algorithms, they are usually much slower in practical application.

Lines 3 – 7

In the for-loop (line 3) the algorithm tests up to $\binom{n}{d}$ ($O(n^d)$) index tuples $i_1, \ldots, i_d$ and calculates

$$\min_R \sum_{k=1}^{d} \left\| q_{i_k} - R\tilde{q}_{j_k} \right\|_2^2$$

for every such index tuple. Here, calculating the optimal orthogonal matrix requires $O(d^3)$ operations. Thus $O(n^d d^3)$ operations are needed for lines 3–7.

Lines 8 – 14

In these lines, two things are calculated. The first is the permutation $\bar{\pi}$ in line 8. Calculating this permutation is essentially equal to finding the nearest neighbour for all $R\tilde{q}_i$ ($i = 1, \ldots, n$) in the set $Q = \{q_1, \ldots, q_n\}$. This *all nearest neighbours* problem can be solved in $O(3^d dn)$ using a cell based algorithm [81].

Besides the permutation $\bar{\pi}$, the corresponding orthogonal transformation $\bar{R}$ is calculated in line 9. Once again, Kabsch's algorithm can be used for this task, adding $O(nd^2 + d^3)$ operations.

One important question remains: how often will these lines actually be called? While the for loop in line 3 may test $O(n^d)$ index tuples, the *continue* statement in line 6 may reduce the number of times that the lines 8 - 14 are called. It is now shown that the lines 8 - 14 will be called no more than $O(2.415^d n^{d-1})$-many times. Thus these lines contribute $O(n^{d-1}2.415^d(3^d dn + nd^2 + d^3)) = O(n^d 7.245^d d)$ operations to the total cost of algorithm 2 if a cell based neighbour search is used to calculate $\bar{\pi}$.

Theorem 8 show that these lines are called only $O(2.415^d n^{d-1})$ times. First, however, a few auxiliary theorems are needed.

Lemma 2. *[82, 83] Let $A \in \mathbb{R}^{d\times d}$ be a square matrix and denote by $A(i)$ the matrix A with its i-th column removed. Then some index $k \in \{1, \ldots, d\}$ exist such that*

$$\left\| A - A(k)A(k)^+ A \right\|_F \leq \sqrt{d}\sigma_d(A).$$

Here, $\sigma_d(A)$ denotes the d-th largest (i.e. smallest) singular value of A and $A(k)^+$ is the Moore-Penrose pseudo-inverse of $A(k)$.

Theorem 7. *Let $A \in \mathbb{R}^{d\times d}$ be non-singular and $k \in \{1, \ldots, d\}$ be an index such that*

$$\left\| A - A(k)A(k)^+ A \right\|_F \leq \sqrt{d}\sigma_d(A)$$

(due to lemma 2 at least one such index exists). Then two vectors $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$ exist such that all orthogonal matrices $U \in \mathbb{O}_d$ fulfil either

$$\left\| x - UA_{*,k} \right\|_2 \leq (1 + \sqrt{2})\sqrt{d}\, \|UA(k) - A(k)\|_F$$

or

$$\left\|y - UA_{*,k}\right\|_2 \leq (1+\sqrt{2})\sqrt{d}\,\|UA(k) - A(k)\|_F\,.$$

Proof. To simplify the notation, it is assumed w.l.o.g. that $k = d$. As A is non-singular, its d-th column $A_{*,d}$ can also be written as

$$A_{*,d} = A(d)\underbrace{A(d)^+ A_{*,d}}_{=:v} + c$$

where $c \neq 0$ is from the (one-dimensional) nullspace of $A(d)^T$. It fulfils $\|c\|_2 \leq \sqrt{d}\sigma_d(A)$ due to

$$\|c\|_2 = \left\|A_{*,d} - A(d)A(d)^+ A_{*,d}\right\|_2 = \left\|(A - A(d)A(d)^+ A)e_d\right\|_2 \leq \sqrt{d}\sigma_d(A).$$

As the smallest singular value of $A(d)$ fulfils $\sigma_{d-1}(A(d)) \geq \sigma_d(A)$ [84], the inequality $\left\|A(d)^+\right\|_2 = 1/\sigma_{d-1}(A(d)) \leq 1/\sigma_d(A)$ holds and therefore

$$\left\|A(d)^+\right\|_2 \|c\|_2 \leq \frac{1}{\sigma_d(A)}\sqrt{d}\sigma_d(A) \leq \sqrt{d}.$$

Furthermore, $\|v\|_2$ fulfils $\|v\|_2 \leq \sqrt{d-1}$ due to

$$\begin{aligned}
\left\|e_d - [I\,0]^T v\right\|_2 &= \left\|Ie_d - A^{-1}A(d)A(d)^+ Ae_d\right\|_2 \\
&= \left\|A^{-1}(Ae_d - A(d)A(d)^+ Ae_d)\right\|_2 \leq \left\|A^{-1}\right\|_2 \left\|A - A(d)A(d)^+ A\right\|_F \left\|e_d\right\|_2 \\
&\leq \frac{1}{\sigma_d(A)}\sqrt{d}\sigma_d(A) = \sqrt{d} \\
\Rightarrow \left\|e_d - [I\,0]^T v\right\|_2^2 &= \left\|\begin{pmatrix} v \\ 1 \end{pmatrix}\right\|_2^2 \leq d \Rightarrow \|v\|_2^2 \leq d - 1.
\end{aligned}$$

It is now shown that $x = A_{*,d} = A(d)v + c$ and $y = A(d)v - c$ fulfil the theorem. Therefore it remains to show that for arbitrary $U \in \mathbb{O}_d$ either

$$\|A(d)v + c - U(A(d)v + c)\|_2 \leq (1+\sqrt{2})\sqrt{d}\,\|UA(d) - A(d)\|_F \tag{4.20}$$

or

$$\|A(d)v - c - U(A(d)v + c)\|_2 \leq (1+\sqrt{2})\sqrt{d}\,\|UA(d) - A(d)\|_F \tag{4.21}$$

holds. For simplicity the definition $\delta = UA(d) - A(d)$ is used. Then $A(d)v - UA(d)v$ can be written as

$$A(d)v - UA(d)v = A(d)v - (A(d) + \delta)v = -\delta v.$$

The next step is more complicated as there is no easy expression for Uc that depends directly on δ. Due to the properties of orthogonal matrices one can easily verify that $\tilde{c} := Uc$ must have the

same Euclidean norm as c and must be in the nullspace of $(UA(d))^T$. Therefore $\tilde{c}$ must fulfil

$$(UA(d))^T \tilde{c} = (A(d) + \delta)^T \tilde{c} = 0 \tag{4.22}$$

$$\tilde{c}^T \tilde{c} = c^T c. \tag{4.23}$$

Using once again the fact that $A(d)$ and c span the whole $\mathbb{R}^d$, $\tilde{c}$ can be written as $\tilde{c} = A(d)w + ac$ for some (yet to be determined) $w \in \mathbb{R}^{d-1}$ and $a \in \mathbb{R}$. Due to equation (4.22), $\tilde{c}$ must fulfil

$$A(d)^T \tilde{c} = -\delta^T \tilde{c}$$

and thus

$$A(d)^T (A(d)w + ac) = A(d)^T A(d)w = -\delta^T \tilde{c}.$$

As $A(d)^T A(d)$ is invertible, w fulfils

$$w = -(A(d)^T A(d))^{-1} \delta^T \tilde{c}.$$

As $\|\tilde{c}\|_2 = \|c\|_2$ due to the second equation, $\|A(d)w\|_2$ fulfils

$$\begin{aligned} \|A(d)w\|_2 &= \left\| -A(d)(A(d)^T A(d))^{-1} \delta^T \tilde{c} \right\|_2 = \left\| -A(d)^{T^+} \delta^T \tilde{c} \right\|_2 \\ &\leq \left\| A(d)^{T^+} \right\|_2 \left\| \delta^T \right\|_F \|\tilde{c}\|_2 = \left\| A(d)^+ \right\|_2 \|\delta\|_F \|c\|_2 . \end{aligned} \tag{4.24}$$

Once again using $\|c\|_2 = \|\tilde{c}\|_2$ one obtains

$$\begin{aligned} c^T c &= w^T A(d)^T A(d)w + a^2 c^T c + 2aw^T \underbrace{A(d)^T c}_{=0} \\ &= \|A(d)w\|_2^2 + a^2 c^T c \end{aligned}$$

which is a quadratic equation with respect to a and thus has two solutions a_+ and a_-, given by

$$a_\pm = \pm \sqrt{1 - \frac{\|A(d)w\|_2^2}{c^T c}}.$$

As $\|A(d)w\|_2$ is limited from above (see equation (4.24)), a_+ fulfils

$$a_+ = \sqrt{1 - \frac{\|A(d)w\|_2^2}{c^T c}} \geq 1 - \frac{\|A(d)w\|_2^2}{c^T c} \geq 1 - \left\| A(d)^+ \right\|_2^2 \|\delta\|_F^2$$

and a_-

$$a_- = -\sqrt{1 - \frac{\|A(d)w\|_2^2}{c^T c}} \leq -\left(1 - \frac{\|A(d)w\|_2^2}{c^T c}\right) \leq -\left(1 - \left\|A(d)^+\right\|_2^2 \|\delta\|_F^2\right).$$

If $\tilde{c} = A(d)w + a_+ c$, one obtains

$$\begin{aligned}\|\tilde{c} - c\|_2^2 &= 2c^T c - \underbrace{c^T A(d)w}_{=0} - 2a_+ c^T c = 2c^T c(1 - a_+) \leq 2c^T c \left\|A(d)^+\right\|_2^2 \|\delta\|_F^2 \\ &\leq 2d \|\delta\|_F^2\end{aligned}$$

and thus equation (4.20) holds

$$\begin{aligned}\|A(d)v + c - U(A(d)v + c)\|_2 &= \|-\delta v + c - \tilde{c}\|_2 \leq \|\delta v\|_2 + \|c - \tilde{c}\|_2 \\ &\leq \|\delta\|_F \|v\|_2 + \sqrt{2d} \|\delta\|_F \leq \|\delta\|_F \sqrt{d} + \sqrt{2d} \|\delta\|_F \\ &= (1 + \sqrt{2})\sqrt{d} \|\delta\|_F .\end{aligned}$$

Similarly, if $\tilde{c} = Aw + a_- c$, one obtains

$$\begin{aligned}\|\tilde{c} + c\|_2^2 &= 2c^T c + 2a_- c^T c = 2c^T c(1 + a_-) \leq 2c^T c \left\|A(d)^+\right\|_2^2 \|\delta\|_F^2 \\ &\leq 2d \|\delta\|_F^2\end{aligned}$$

and equation (4.21) is satisfied:

$$\begin{aligned}\|A(d)v - c - U(A(d)v + c)\|_2 &= \|-\delta v - c - \tilde{c}\|_2 \leq \|\delta v\|_2 + \|c + \tilde{c}\|_2 \\ &\leq \|\delta\|_F \|v\|_2 + \sqrt{2d} \|\delta\|_F \leq \|\delta\|_F \sqrt{d} + \sqrt{2d} \|\delta\|_F \\ &= (1 + \sqrt{2})\sqrt{d} \|\delta\|_F .\end{aligned}$$

□

Theorem 8. *Let $(q_1, \ldots, q_n)$ ($q_i \in \mathbb{R}^d$) be n d-dimensional points such that*

$$\left\|q_i - q_j\right\|_2 \geq \mu,\ i \neq j$$

for some $\mu > 0$. Furthermore let $\tilde{q}_1, \ldots, \tilde{q}_d$ ($\tilde{q}_i \in \mathbb{R}^d$) be d d-dimensional points that are linearly independent. Then the magnitude of the set

$$I = \left\{ i \in \{1, \ldots, n\}^d : \left(i_j \neq i_k \ \forall j \neq k\right) \wedge \left(\exists R \in \mathbb{O}_d : \sum_{l=1}^{d} \left\|q_{i_l} - R\tilde{q}_l\right\|_2^2 \leq \epsilon^2\right)\right\}$$

is of order $O(2.415^d n^{d-1})$ *if* $\epsilon < \frac{\mu}{2\sqrt{1+4d}}$.

Proof. To begin with, let $\tilde{Q} = [\tilde{q}_1, \ldots, \tilde{q}_n]$ be the matrix that has the vectors $\tilde{q}_i$ as columns. Then $\tilde{Q}$ fulfils all conditions of lemma 2 and therefore some index k exists such that

$$\left\|\tilde{Q} - \tilde{Q}(k)\tilde{Q}(k)^+\tilde{Q}\right\|_F \leq \sqrt{d}\sigma_d(\tilde{Q}).$$

W.l.o.g. it is assumed here that $k = d$, otherwise one can simply permute the $\tilde{q}_i$ and q_i. Then every $i \in I$ can be written as $i = (j, l)$ where $j \in \{1, \ldots, n\}^{d-1}$ and $l \in \{1, \ldots, n\}$. If $(j, l) \in I$, j must fulfil

$$\left(j_m \neq j_p \; \forall m \neq p\right) \wedge \left(\exists R \in \mathbb{O}_d : \sum_{m=1}^{d-1} \left\|q_{j_m} - R\tilde{q}_m\right\|_2^2 \leq \epsilon^2\right).$$

Due to the condition that all entries of j must be distinct, there are only up to $\binom{n}{d-1}$ possible tuples j. It is now shown that for each such j, no more than $2 \cdot 2.415^d$ possible values $\tilde{l}$ exist such that $(j, \tilde{l}) \in I$.

To prove this, it can easily be observed that for $i = (j, l) \in I$ the relation

$$\underbrace{\left\{R \in \mathbb{O}_d : \sum_{m=1}^{d-1} \left\|q_{j_m} - R\tilde{q}_m\right\|_2^2 \leq \epsilon^2\right\}}_{=\mathcal{R}_j} \supseteq$$
$$\left\{R \in \mathbb{O}_d : \sum_{m=1}^{d-1} \left\|q_{j_m} - R\tilde{q}_m\right\|_2^2 + \left\|q_l - R\tilde{q}_d\right\|_2^2 \leq \epsilon^2\right\}$$

holds. Furthermore, all $R, \tilde{R} \in \mathcal{R}_j$ fulfil

$$\left\|R\tilde{Q}(d) - \tilde{R}\tilde{Q}(d)\right\|_F \leq 2\epsilon. \tag{4.25}$$

Therefore, choosing an arbitrary $R \in \mathcal{R}_j$ and setting $A = R\tilde{Q}$ in theorem 7 yields two vectors $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$ such that $\forall \tilde{R} \in \mathcal{R}_j$ either x or y fulfil

$$\begin{aligned}\left\|x - \tilde{R}\tilde{Q}_{*,d}\right\|_2 &= \left\|x - \tilde{R}R^T(R\tilde{Q}_{*,d})\right\|_2 \leq (1+\sqrt{2})\sqrt{d}\left\|R\tilde{Q}(d) - \tilde{R}R^T(R\tilde{Q}(d))\right\|_F \\ &= (1+\sqrt{2})\sqrt{d}\left\|R\tilde{Q}(d) - \tilde{R}\tilde{Q}(d)\right\|_F.\end{aligned}$$

Due to equation (4.25) this leads to either

$$\left\|x - \tilde{R}\tilde{q}_d\right\|_2 \leq 2(1+\sqrt{2})\sqrt{d}\epsilon$$

or

$$\left\|y - \tilde{R}\tilde{q}_d\right\|_2 \leq 2(1+\sqrt{2})\sqrt{d}\epsilon.$$

Due to these inequalities, each feasible $\tilde{R} \in \mathcal{R}_j$ transforms $\tilde{q}_d$ into a sphere of radius $2(1+\sqrt{2})\sqrt{d}\epsilon$ around either x or y. Thus, the corresponding q_d must also lie in one of these spheres. As the q_i were assumed to fulfil $\left\|q_i - q_j\right\|_2 \geq \mu$, only a limited number of q_i-s can lie in these spheres. As the diameter of each sphere is no more than

$$\frac{2 \cdot 2(1+\sqrt{2})\sqrt{d}\epsilon}{\mu} < \frac{2 \cdot 2(1+\sqrt{2})\sqrt{d}\epsilon}{2\sqrt{1+4d\epsilon}} = \frac{2(1+\sqrt{2})\sqrt{d}}{\sqrt{d}\sqrt{\frac{1}{d}+4}} \leq \frac{2+2\sqrt{2}}{\sqrt{4}} \leq 2.415$$

times larger than μ – and thus independent of n – the maximal number of points q_i that fit in such a sphere cannot be larger than 2.415^d.

Thus if $(j,l) \in I$, for each j there are no more than $2 \cdot 2.415^d$ possible $\tilde{l} \in \{1,\ldots,n\}$ such that $(j,\tilde{l})$ is also in I. □

Comment 1. *The previous theorem shows that the magnitude of the set I is of order $O(2.415^d n^{d-1})$. The conducted numerical experiments suggest that the actual bound on the magnitude may be as low as $O(n^{d-1})$ as I was not able to find a case where for a fixed j there are more than two choices for l such that $(j,l) \in I$.*

Using the complexities of all components of the algorithm, the computational complexity of the whole algorithm can be determined. As shown in table 4.1, the final complexity is $O(n^d 7.245^d d)$ if a cell-based neighbour search is used. As atoms are usually described by their two- or three-dimensional coordinates, the dimensions $d = 2$ and $d = 3$ are of special interest here. In these cases, algorithm 2 scales as $O(n^2)$ $(d = 2)$ or $O(n^3)$ $(d = 3)$ with n being the number of atoms in a neighbourhood.

Line 2 (optimal indices)	$O(n^d d^3)$
Lines 3–7	$O(n^d d^3)$
Lines 8 – 14	$O(n^d 7.245^d d)$
Total	$O(n^d 7.245^d d)$

Table 4.1: Components of the complexity analysis.

4.6 The local transition database

Now that the problem of comparing neighbourhoods has been tackled, an efficient database to store atomic neighbourhoods and corresponding local transitions should be designed.

Two important questions must be answered: *How can one find similar neighbourhoods and corresponding data in an efficient way?* and *How can one represent the local transitions?*

4.6.1 Database structure

In the original k-ART algorithm, the process of finding similar neighbourhoods was possible in a very efficient manner. First, the neighbourhood was converted to the corresponding adjacency graph. Afterwards, nauty was used to calculate the canonical labelling of this graph. By construction, neighbourhoods with the same canonical labelling were seen as equivalent. Thus the database could easily be realised as a simple key/value map where the key is the canonical labelling of the graph (which can be represented as a simple string). Finding a neighbourhood N in the k-ART database therefore required just two steps: the calculation of the canonical label of N and a single look-up operation in a map with string-keys.

When the (fully) invariant RMSD is used to compare atomic environments, this simple key-based structure can no longer be used. In order to avoid looking at every single neighbourhood in the database, some filtering mechanisms are needed to limit the number of necessary comparison operations.

Element filtering

The first idea for a filter is straightforward. If two neighbourhoods are similar with respect to the invariant RMSD, their elements must match. Thus one can ignore all neighbourhoods in the database that do not have the same elements as the neighbourhood N.

To do this, an *element identifier* is calculated for each neighbourhood. The element identifiers of two neighbourhoods should be equal if they have the same elements and different if they are not. Possible choices are the sorted list of elements in the neighbourhood or a compressed representation of it, for example "C2H6" for a neighbourhood with two carbon and six hydrogen atoms. The full local transition database is then realised as a map of several ⟨element identifier, subdatabase⟩ pairs where each subdatabase holds neighbourhoods that share a common element identifier.

Range searching

The element filtering approach limits the database query to neighbourhoods that have matching elements (and thus the same number of particles). To limit the number of comparison operations further, another filter can be built, using the particle distances in neighbourhoods. The basis for this is the following theorem that offers a necessary condition for $d^{full}(N, \tilde{N}) \leq \epsilon$.

Theorem 9. *Let $N = (q, e)$ and $\tilde{N} = (\tilde{q}, \tilde{e})$ be two neighbourhoods. If $d^{full}(N, \tilde{N}) \leq \epsilon$ there*

exists a permutation π such that

$$\left| \left\|q_i\right\|_2 - \left\|\tilde{q}_{\pi_i}\right\|_2 \right| \leq \epsilon \qquad i = 1, \ldots, n \tag{4.26}$$

$$e = \tilde{e}_\pi. \tag{4.27}$$

Proof. If $d^{full}(N, \tilde{N}) \leq \epsilon$ let R^* and π^* be the corresponding optimal orthogonal transformation and permutation. Obviously they fulfil $e = \tilde{e}_{\pi^*}$ and $g(N, \tilde{N}, R^*, \pi^*) \leq \epsilon^2$. Thus one can write

$$\begin{aligned}
\epsilon^2 \geq g(N, \tilde{N}, R^*, \pi^*) &= \sum_{k=1}^{n} \left\| q_i - R^* \tilde{q}_{\pi_i^*} \right\|_2^2 \\
&\geq \sum_{i=1}^{n} \left| \left\|q_i\right\|_2 - \left\|R^* \tilde{q}_{\pi_i^*}\right\|_2 \right|^2 \\
&= \sum_{i=1}^{n} \left| \left\|q_i\right\|_2 - \left\|\tilde{q}_{\pi_i^*}\right\|_2 \right|^2 \\
&\geq \left| \left\|q_k\right\|_2 - \left\|\tilde{q}_{\pi_k^*}\right\|_2 \right|^2 \qquad \forall k = 1, \ldots, n.
\end{aligned}$$

□

Theorem 9 can now be used to filter the neighbourhoods in the database. For a given neighbourhood $N = (q, e)$ a *distance identifier* of the form

$$\Phi(N) = \left[\left\|q_{\pi_1}\right\|_2, \ldots, \left\|q_{\pi_n}\right\|_2 \right]$$

can be built. π is the permutation that corresponds to sorting the particles in N such that

$$\begin{aligned}
e_{\pi_i} &\leq e_{\pi_{i+1}} && i = 1, \ldots, n-1 \\
\left\|q_{\pi_i}\right\|_2 &\leq \left\|q_{\pi_{i+1}}\right\|_2 && \text{if } e_{\pi_i} = e_{\pi_{i+1}}
\end{aligned}$$

(assuming that there is some natural order for the elements). For two neighbourhoods N and $\tilde{N}$ it can be shown that if

$$\left\|\Phi(N) - \Phi(\tilde{N})\right\|_\infty > \epsilon,$$

there exists no permutation $\tilde{\pi}$ that fulfils equations (4.26) and (4.27). Using theorem 9 this implies that $d^{full}(N, \tilde{N}) > \epsilon$. Thus one only has to consider neighbourhoods $\tilde{N}$ in the database that fulfil

$$\left\|\Phi(N) - \Phi(\tilde{N})\right\|_\infty \leq \epsilon.$$

The problem of finding all these neighbourhoods $\tilde{N}$ corresponds to the *orthogonal range*

searching problem.

Definition 9. *Let $X = \{x_1, \ldots, x_n\}$ be a set of n d-dimensional points. If intervals $[a_j, b_j]$ $(j = 1, \ldots, d)$ are given, the problem of finding all points $x_i \in X$, such that*

$$(x_i)_j \in [a_j, b_j], \quad j = 1, \ldots, d \tag{4.28}$$

is called the orthogonal range searching *problem.*

Special data structures and algorithms for the orthogonal range searching problem exist. In [85] a range tree was presented that supports adding and removing points. For a set $X = \{x_1, \ldots, x_n\}$ of n d-dimensional points, it can find all $x_i \in X$ that fulfil equation (4.28) in $O(k + \log^{d-1}(n) \log(\log(n)))$ where k is the number of points that actually fulfil this condition. If adding and removing points is not necessary, the query time can be lowered to $O(k + \log^{d-1}(n))$ [86].

Using these two filtering stages, the local transition database has the structure that is shown in figure 4.7. Neighbourhoods with different elements are stored in different subdatabases. Each subdatabase is essentially realised as a range tree that holds neighbourhoods, their corresponding distance identifiers and local transition data.

4.6.2 Storing local transitions

Whenever a new local transition has been found, it should be added to the corresponding neighbourhood in the database. To store a local transition, the initial neighbourhood is written to the database. In order to describe the structural changes in this neighbourhood during a local transition the *saddle shift* and *product shift* are also stored. The saddle shift $(s_1 - r_1, \ldots, s_n - r_n)$ is the difference between the particle coordinates r_i in the initial neighbourhood and the coordinates s_i at the saddle point. The product shift $(p_1 - r_1, \ldots, p_n - r_n)$ is defined in a similar way. Here, the p_i denote the coordinates at the end of the transition.

Besides these shifts in the coordinates, additional information related to HTST is stored. This includes the energy barriers in both directions as well as the corresponding HTST prefactors.

4.6.3 Retrieving local transitions

So how can local transitions that may take place inside some neighbourhood $N = (q, e)$ be retrieved from the database? The first step is to calculate the element identifier of the neighbourhood as well as the distance identifier $\Phi(N)$. Using the element identifier, the right range tree in the database can be chosen (see figure 4.7). It can then be used to find all neighbourhoods $\tilde{N}_1, \ldots, \tilde{N}_k$ in the database that have similar distance identifiers.

In the next step, the check $d^{full}(N, \tilde{N}_i) \leq \epsilon$ is run for all these neighbourhoods. If the check fails for all $\tilde{N}_i$, no matching local transitions for N are stored in the database. Otherwise, the neighbourhood $\tilde{N}_i$ with the smallest distance $d^{full}(N, \tilde{N}_i)$ is chosen and is denoted by $\tilde{N}^*$. Using

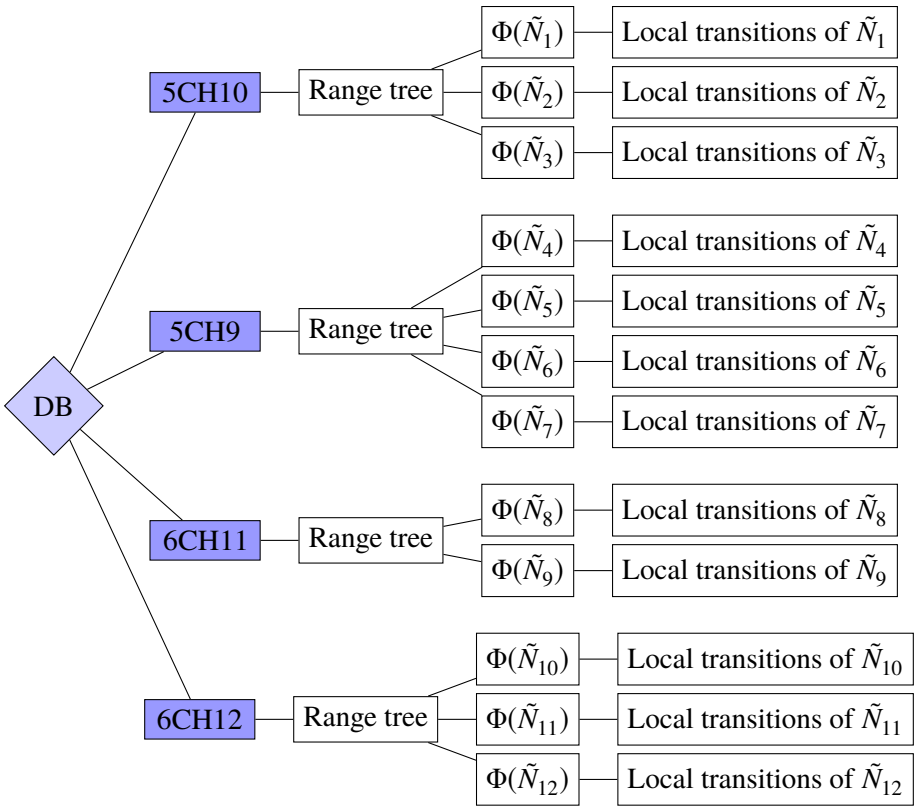


Figure 4.7: Structure of the local transition database.

the optimal transformations R^* and π^*, the local transitions that belong to $\tilde{N}^*$ can finally be transformed such that they can be applied to N instead of $\tilde{N}^*$.

4.7 The Localised Kinetic Monte Carlo algorithm

The new invariant RMSD comparison function and the corresponding local transition database are now used to build a modified version of k-ART that is called the *Localised Kinetic Monte Carlo (LKMC)* algorithm. LKMC has the same basic structure as k-ART (see figure 4.4) but differs from it in various ways. Some modifications were made in order to improve the performance or robustness of LKMC, others were made purely for technical reasons. LKMC has been implemented as an extension module for the QuantumATK software package [87, 88]. As QuantumATK already includes a (non-localised) on-the-fly KMC algorithm called *adaptive KMC (AKMC)*, some of its functionality – like global saddle searches – is used, instead of reimplementing the corresponding k-ART analogue. The modifications are now presented in

more detail.

4.7.1 Local transition searches

In the original k-ART algorithm, local transition searches were carried out by using the the ART nouveau method (which also gave k-ART its name) on each distinct neighbourhood [42]. For LKMC a different approach was chosen that combines the neighbourhood-wise local transition searches in k-ART with the active volume approach in [49]. To explain the basic concept, it is temporarily assumed that local transition searches are needed only for a single atomic neighbourhood with radius r_{cut} in the larger global system. In this case, the local transition search approach from [52] can be used: all particles in the global system that correspond to the particles in the neighbourhood are allowed to move, while all other particles are kept fixed. If the chosen potential energy function E is short-ranged with interaction radius $r_{interact}$ [6], all particles whose Euclidean distance to the central particle of the neighbourhood is larger than $r_{cut} + r_{interact}$ can be ignored (see figure 4.8).

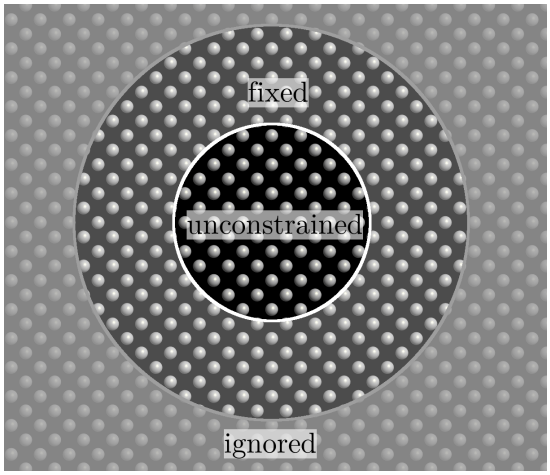


Figure 4.8: Setup of a local transition search for a single neighbourhood.

Standard global transition searches are then applied to this smaller and constrained system. As LKMC builds on top of the AKMC implementation in QuantumATK, the QuantumATK transition search functionality is used, which is a three-step procedure [89]. First, a high-temperature trajectory that starts in the initial potential energy minimum is calculated and analysed in order to detect whether the trajectory escapes from the initial energy basin. If it does, a nudged elastic band approach is used to find the reaction path that connects the two

[6] Here, E being short-ranged means that $\frac{\partial^2 E(x)}{\partial x_i \partial x_j} = 0$ whenever $\left\| x_i - x_j \right\|_2 > r_{interact}$. If E is a sum of pair potentials this would be the case if all pair potentials are truncated at $r_{interact}$.

energy basins. Using the reaction path, the saddle point, energy barrier, and harmonic prefactor of the transition are calculated. This procedure is repeated until all neighbouring states have been found (see also subsection 4.7.2).

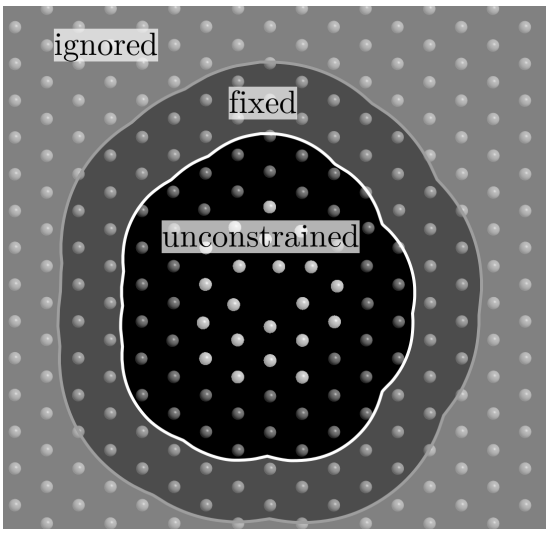


Figure 4.9: Setup of local transition searches in multiple neighbourhoods.

In general, the local transitions for more than one atomic neighbourhood are unknown. As in the single-neighbourhood case, a smaller constrained system is built and standard transition searches are applied to it. Figure 4.9 shows how the local transitions in the neighbourhoods of the bright particles can be calculated. Once again, all particles in the global system that correspond to a particle in one of the neighbourhoods are allowed to move while all other particles are kept fixed. Furthermore, all particles are ignored whose Euclidean distance to all central particles of the neighbourhoods is larger than $r_{cut} + r_{interact}$. Using this approach, LKMC basically builds non-spherical active regions on the fly, using the stored local transition information.

One obvious consequence of this approach is that the transitions that occur in this smaller constrained system must be mapped back to the individual neighbourhoods. This is done using the concept of *process atoms* in [51]. For each transition that takes place in the constrained system, all atoms that move more than some given distance $d_{transition}$ during this transition are called *process atoms*. Then, a transition in the constrained system is assigned to one of the neighbourhoods if the process atoms stay inside this neighbourhood for the whole transition.

4.7.2 Confidence estimation

In the previous section it was briefly noted that transition searches in the constrained local system should be run "until all neighbouring states have been found". Therefore some kind of stopping criterion is needed in order to terminate the transition searches at some point. In k-ART no such stopping criterion is used. Instead, a fixed number of transition searches are conducted for each distinct neighbourhood [64]. This number is set by hand and must be large enough such that all necessary transitions are found and small enough such that the transition searches do not become excessively expensive.

In LKMC, the confidence estimator from [41] is used, which estimates the probability that all necessary transitions have been found. If the confidence becomes larger than some given tolerance, the transition searches are stopped. This estimator requires that the transition searches use high temperature MD trajectories to escape from the initial energy minimum, which is the case in the LKMC algorithm.

A potential issue with this estimator is that it assumes that some (accessible) transitions exist. In LKMC this is not always the case, for example when LKMC tries to determine local transitions in a neighbourhood with a perfect crystal structure. Without any modifications, the transition searches would never terminate. Therefore, an additional stopping criterion is used in LKMC. If the high-temperature trajectory that is calculated during the transition searches does not manage to escape from the current state after some time t_{max}, the transition search is stopped.

4.7.3 Recycling local transitions

The final component of the LKMC algorithm is the recycling of known local transitions. For each atom in a given particle system, the corresponding local transitions are retrieved from the LKMC database. As these local transitions are stored as shifts of the particle positions in this neighbourhood, they can immediately be applied to the atoms in the neighbourhood, thus obtaining good estimates for the global saddle and product configuration.

Afterwards, the exact global product configuration is calculated by relaxing the estimated product configuration towards its actual energy minimum. The exact saddle point is then calculated in the same way, using the global saddle search mechanism with the estimated saddle point as its initial value. This procedure is then repeated for all other neighbourhoods in the atomistic system until all local transitions have been reused.

CHAPTER 5

Evaluation

The different ideas that were presented in the previous chapters are now tested in a series of computational experiments. As was briefly noted before, the proposed LKMC algorithm as well as all necessary components have been implemented as extension modules for the QuantumATK software package [87, 88].

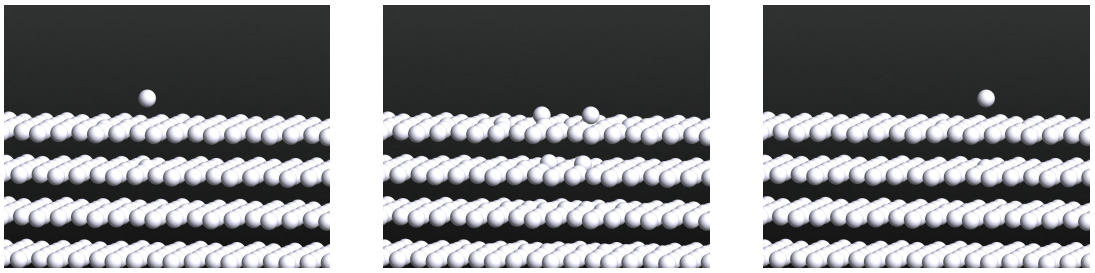
5.1 Example problems

During the evaluation, a number of test problems are used repeatedly to investigate certain properties of the different components of LKMC. These test problems are now presented in more detail.

5.1.1 Ad-atom hopping on a platinum surface

The first example problem is the movement of platinum ad-atoms on a [001] platinum surface. Up to a few decades ago, it was thought that an ad-atom simply hops from one place on the surface to another. Kellogg and Feibelman demonstrated in 1990 [90] that the primary reaction mechanism on such a platinum surface is actually an exchange process where the ad-atom dips into the surface and pushes another atom out, as shown in figure 5.1. Using a suitable EAM-potential [91] and the on-the-fly KMC algorithm in QuantumATK (AKMC), this behaviour can be reproduced. As expected, these simulations show that there are two possible mechanisms for the movement of the ad-atom. The first one is the aforementioned exchange process, the other is the naive *hopping process* where the ad-atom simply moves to a neighbouring place on the surface. As the exchange-process has a lower energy barrier of about 0.5 eV compared to the hopping process (0.93 eV), it is the predominant process at moderate temperatures.

Various variants of this ad-atom problem are used. They differ in the size of the surface as well as the number of ad-atoms that are placed onto it. If more than one ad-atom is used, these



(a) Initial state.

(b) Ad-atom pushes another atom out of the surface.

(c) Final state - the previous ad-atom is now a part of the surface.

Figure 5.1: The exchange mechanism on a platinum surface.

ad-atoms may merge to form *islands*.

5.1.2 Hydrogen diffusion in amorphous silicon

One of the main reasons for the development of the new neighbourhood comparison procedure was the desire to use a k-ART like procedure on amorphous systems. Therefore, the diffusion of hydrogen atoms in an amorphous silicon crystal is investigated. Hydrogenated amorphous silicon is used as a semiconductor in different applications [92]. Depending on the temperature, the incorporated hydrogen atoms show a varying degree of diffusivity and may evaporate at high temperatures.

Creating a realistic amorphous system can be difficult, therefore the 4096-particle system from [93] which showed good agreement with experiments is used. Varying numbers of hydrogen atoms are inserted into this system and a long Langevin trajectory is calculated in order to relax the system initially. Afterwards the hydrogen diffusion can be studied using the potential from [94].

5.1.3 Nickel grain boundary

Many materials are not made up of a single perfect crystal structure but are polycrystalline, which means that they consist of many smaller perfect crystals. The interface between two such perfect crystals is called the *grain boundary*. Understanding its evolution is of interest as the grain boundary may move or even vanish, causing the growth or disappearance of the crystals that form a polycrystalline material.

Here, the grain boundary between two nickel crystals is simulated (see figure 5.2). The EAM potential in [91] was used to model the interactions between the nickel atoms.

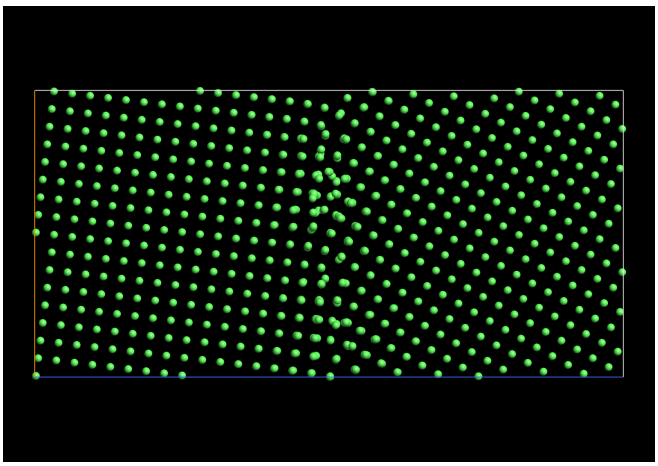


Figure 5.2: Initial state of the grain boundary simulation.

5.2 Localisation of transitions

The first experiments deal with the question of *localisability*, i.e. when can transitions be localised, what cutoff radii are needed and how do saddle points and minima converge with respect to the cutoff radius.

To investigate the localisability of global transitions, a number of such transitions are analysed. Each global transition is defined by the particle positions in the initial (reactant) state $r \in \mathbb{R}^{n\times 3}$, at the saddle point $s \in \mathbb{R}^{n\times 3}$ and in the final (product) state $p \in \mathbb{R}^{n\times 3}$. As discussed in 4.1, both r and p are local minimisers of the potential energy function E while s is a first-order saddle point. Then the localisability is analysed using the following procedure.

1. The initial position of the particle that moves the most during the global transition is determined and is used as the centre of the transition.

2. Using the centre of the transition and different cutoff radii r_{cut}, different spherical active regions $S(r_{cut})$ are created. Using the ideas in section 4.1 and subsection 4.7.1, they are used to create the corresponding localised problems by constraining all particles that do not lie inside the active region.

3. The inbuilt optimisation algorithm (L-BFGS) in QuantumATK is used to find the local minimiser $\tilde{p}(r_{cut}) \in \mathbb{R}^{n\times 3}$ of the localised problem with cutoff r_{cut} that is closest to p. Similarly, the first order saddle point $\tilde{s}(r_{cut}) \in \mathbb{R}^{n\times 3}$ that is closest to s is determined.

4. The errors $\left\|p - \tilde{p}(r_{cut})\right\|_F$ and $\left\|s - \tilde{s}(r_{cut})\right\|_F$ are determined.

5. Finally, it is tested whether the inbuilt optimiser in QuantumATK (L-BFGS) recovers the

product configuration p if $\tilde{p}(r_{cut})$ is used as the initial guess. Similarly, $\tilde{s}(r_{cut})$ is used as the initial guess for finding the global saddle point.

Platinum ad-atom – exchange process The first global transition that is investigated is the exchange process of a platinum ad-atom as explained in 5.1.1. In this experiment, a single platinum ad-atom is placed on a 111 Å × 111 Å [001] platinum surface.

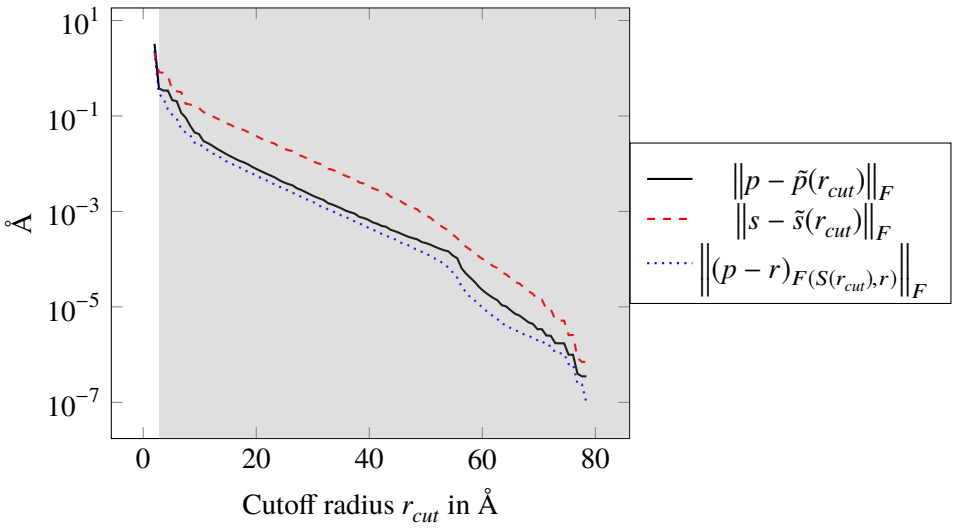
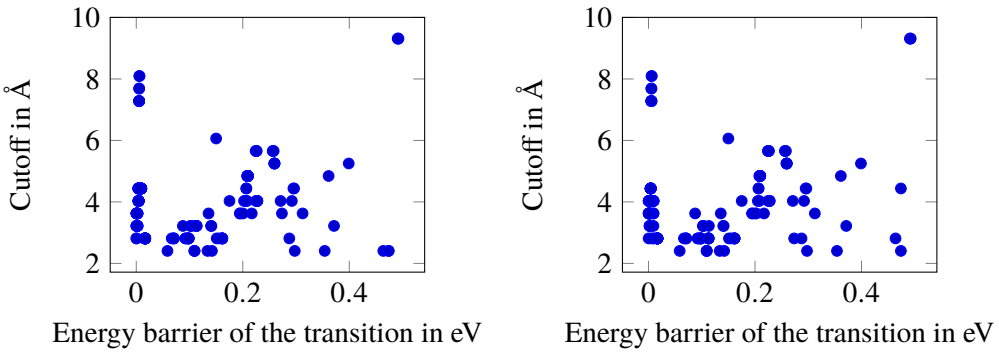


Figure 5.3: Platinum ad-atom: difference in local and global solutions.

Figure 5.3 shows three different curves. The first one shows the difference in the coordinates $(\|(p - r)_{F(S(r_{cut}),r)}\|_F)$ between the global reactant state r and the global product state p, limited to the particles that are kept fixed for some given cutoff radius r_{cut}. The other two represent the difference between the global product/saddle configurations and their localised approximations $(\|p - \tilde{p}(r_{cut})\|_F, \|s - \tilde{s}(r_{cut})\|_F)$.

In the figure, it is shown that the localised product states converge towards the global product state in a very similar manner as the difference in the fixed coordinates. This is in agreement with the Lipschitz bound on the error that was presented in section 4.1. At around $r_{cut} = 2.8$ Å, a large drop in all three curves is visible. This does not come as a surprise, as the distance between neighbouring platinum atoms is about 2.8 Å. Therefore, using a smaller cutoff radius in the localisation will keep all particles except the ad-atom fixed, which of course prohibits the exchange mechanism. The grey area in figure 5.3 denotes all cutoff radii, such that using $\tilde{p}(r_{cut})$ as the initial guess in the optimiser recovers p. As shown, this is the case for all cutoff radii larger than 2.8 Å.

Amorphous silicon The next test is conducted on the amorphous silicon example in subsection 5.1.2 where five additional hydrogen atoms have been added at random positions. Due to the amorphous structure, no predominant transition mechanism exists, but a high number of different mechanisms with similar importance. Therefore, the AKMC algorithm in QuantumATK is used to simulate the amorphous system over 10 KMC steps. All global transitions that are encountered in this simulation are stored and each of them is analysed as before. Figure 5.4



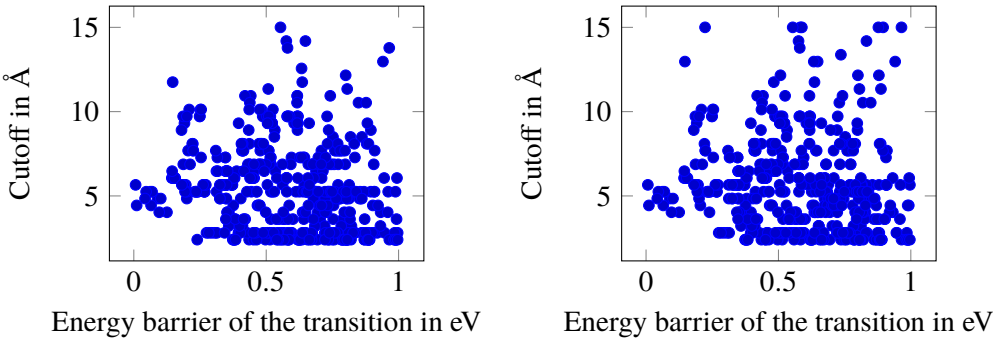
(a) Necessary cutoffs to recover the products.

(b) Necessary cutoffs to recover the saddles.

Figure 5.4: Cutoff study for the amorphous silicon problem.

shows for each transition the cutoff radii that are required to recover the transition products and saddles. It can be observed, that no obvious relation between the size of the energy barrier and the necessary cutoff seems to exist. Furthermore, most transitions require a cutoff radius that is well below 8 Å, suggesting that they can be localised.

Grain boundary To analyse the localisability of the grain boundary problem in 5.1.3, the same approach as for the amorphous silicon problem is used. By analysing all encountered global transitions, the necessary cutoffs for all these transitions were determined and can be found in figure 5.5. Many transitions require rather large cutoffs in order to be properly captured. This is not only visible in figure 5.5 but can also be verified by looking at the actual global transitions, which can affect a large number of particles that are not necessarily close to each other.



(a) Necessary cutoffs to recover the products.

(b) Necessary cutoffs to recover the saddles.

Figure 5.5: Cutoff study for the grain boundary problem.

5.3 Comparing atomic neighbourhoods

One of the central contributions of this thesis is the new algorithm 2 that can be used to test neighbourhoods N and $\tilde{N}$ for similarity. This algorithm is now compared to other similar algorithms to evaluate its accuracy and performance. To test whether invariance under both orthogonal transformations and permutations is really necessary, a number of additional tests are conducted in subsection 5.3.5.

As reference algorithms Go-ICP [76] and Go-Permdist [69] were chosen. The reason for that choice is that both algorithms are exact and deterministic and could thus be used for the comparison routine in the LKMC database. Another important reason is that optimised reference implementations are available. Whereas Go-ICP originates from computer vision, Go-Permdist was designed to align particle system. Both algorithms are branch-and-bound based and are able to calculate the invariant RMSD $d^{full}(N, \tilde{N})$ up to some tolerance δ if the optimal orthogonal transformation is a rotation matrix (i.e. has determinant 1). For both Go-ICP and Go-Permdist the implementation of the authors [95, 96] is used with the following modifications:

- Both algorithms calculate an invariant RMSD that is invariant not only under permutations and rotations but also under translations. As neighbourhoods have a fixed centre, no translational invariance is needed. Therefore, the calculation of the optimal translation in both algorithms was disabled.

- The two codes were modified such that the calculation stops as soon as the lower bound in the branch-and-bound part of the algorithms becomes larger than the comparison tolerance ϵ. The rationale behind this was that if $d^{full}(N, \tilde{N}) > \epsilon$, the actual value of $d^{full}(N, \tilde{N})$ is of no interest, only the fact that the two neighbourhoods are not similar.

Thus it is sufficient to stop the branch-and-bound parts as soon as it is established that $d^{full}(N, \tilde{N}) > \epsilon$.

- As a stopping criterion for the branch-and-bound parts of these algorithms $\delta = 1 \times 10^{-2}$ Å was used, i.e. the algorithms were stopped as soon as the difference between lower bound for $d^{full}(N, \tilde{N})$ and the corresponding upper bound became smaller than δ.

- During the calculations it became apparent that the Go-Permdist algorithm sometimes calculated wrong results. The cause of this problem turned out to be the lower bound that is used in Go-Permdist [69, equation 19], which is not correct in all cases. Thus I replaced the faulty lower bound by a corrected version (see appendix A.1 for a detailed explanation).

A number of test datasets are also needed to evaluate the algorithms. The following datasets were used:

Silicon dataset This dataset was created using the collection of silicon configurations in [97,98]. Each configuration was decomposed into atomic neighbourhoods using the cutoff radius 6 Å. Out of these neighbourhoods, one million neighbourhood pairs $(N_i, \tilde{N}_i)$ were randomly chosen such that N_i and $\tilde{N}_i$ have the same number of particles.

Amorphous dataset Using the AKMC algorithm in QuantumATK, the initial amorphous silicon system in subsection 5.1.2 without any hydrogen atoms was simulated over ten KMC steps[1]. The ten different configurations that were encountered during this simulation were decomposed into neighbourhoods, using 6 Å as the cutoff radius. As in the silicon dataset, one million random neighbourhood pairs $(N_i, \tilde{N}_i)$ that have the same number of particles were chosen.

Diamond dataset For different cutoff radii r, the atomic neighbourhood $N^{(r)} = (q^{(r)}, e^{(r)})$ of a particle in a perfect diamond crystal was calculated[2]. Using the coordinates and elements of this neighbourhood, 1000 neighbourhood pairs $(N_i^{(r)}, \tilde{N}_i^{(r)})$ were created using the formula

$$N_i^{(r)} = (R^{(i)} q^{(r)}_{\pi^{(i)}}, e^{(r)}_{\pi^{(i)}}) \tag{5.1}$$

$$\tilde{N}_i^{(r)} = (\tilde{R}^{(i)} q^{(r)}_{\tilde{\pi}^{(i)}} + \frac{\tau^{(i)}}{\left\|\tau^{(i)}\right\|_F} \rho^{(i)}, \tilde{e}^{(r)}_{\tilde{\pi}^{(i)}}). \tag{5.2}$$

[1] As Go-ICP cannot deal with different elements, only systems where all particles have the same element were tested here.

[2] All particles in a perfect diamond crystal have the same neighbourhood, therefore the choice of the particle does not matter.

$R^{(i)}$ and $\tilde{R}^{(i)}$ are random 3×3 rotation matrices whereas $\pi^{(i)}$ and $\tilde{\pi}^{(i)}$ are random permutations. Additional noise was added in the form of $\tau^{(i)}$ which is distributed according to the multivariate normal distribution. An additional factor $\rho^{(i)}$, drawn from the exponential distribution with parameter $\lambda = 3$, was added such that the magnitude of the noise $\left\| \tau^{(i)} / \|\tau^{(i)}\|_F \rho^{(i)} \right\|_F$ is exponentially distributed and thus covers cases where $N_i^{(r)}$ and $\tilde{N}_i^{(r)}$ are very similar ($\rho^{(i)}$ small) and where they are not ($\rho^{(i)}$ large).

Sphere dataset The last dataset was specifically designed to be as challenging for algorithm 2 as possible. For different numbers of particles n, the algorithm in [99] was used to distribute $n - 1$ points in an almost uniform manner on the surface of the unit sphere. An additional point with coordinates $(0, 0, 0)$ was also added. The resulting points were then scaled such that the minimum distance between two particles was 2 Å. Using these scaled coordinates $q^{(n)}$ and n equal elements $e_1^{(n)} = e_2^{(n)} = \cdots = e_n^{(n)}$, 100 neighbourhood pairs $(N_i^{(n)}, \tilde{N}_i^{(n)})$ were built using the same approach as in equation (5.1) but without any noise.

The challenge of this dataset is the high order of symmetry of the particles and the additional problem that all particles (except for the central particle) have exactly the same distance to the origin.

5.3.1 Accuracy

As a first step, the correctness of the algorithm 2 had to be verified. For small neighbourhoods N and $\tilde{N}$ the exact value $d^{full}(N, \tilde{N})$ can be calculated by an exhaustive search over all possible permutations. The corresponding optimal orthogonal transformations can then be obtained by the Kabsch algorithm. This brute-force approach was used to exactly calculate d^{full} for all neighbourhood pairs in the datasets with less than nine particles. Using these brute-force results as references, it could be verified that algorithm 2 calculates the same results on these neighbourhood pairs.

Whenever the neighbourhoods contain more than a few particles, it is no longer possible to calculate d^{full} by a brute-force search. Instead, algorithm 2 was tested against the results of the Go-ICP and Go-Permdist algorithms. Using the silicon dataset, algorithm 2 was in some cases able to find lower d^{full} values than Go-ICP and Go-Permdist. This was expected, as algorithm 2 minimises over all orthogonal transformations, while Go-ICP and Go-Permdist minimise only over all rotations (orthogonal matrices with determinant 1). Thus, algorithm 2 can find better d^{full} values if the optimal orthogonal matrix in the calculation of d^{full} is not a rotation matrix. On the other datasets, all three algorithms produced the same results, up to the specified tolerance (1×10^{-2} Å) that was used as the stopping criterion for both Go-ICP and Go-Permdist.

5.3.2 Performance evaluation

The next step was to investigate the performance of the algorithms in terms of the computational time that is needed to compare two atomic neighbourhoods. To do this, the three algorithms were used to compare all neighbourhood pairs in the datasets. All calculations were run in serial on an Intel Xeon Gold 5118 processor and the tolerance ϵ was set to 0.2 Å unless specified otherwise.

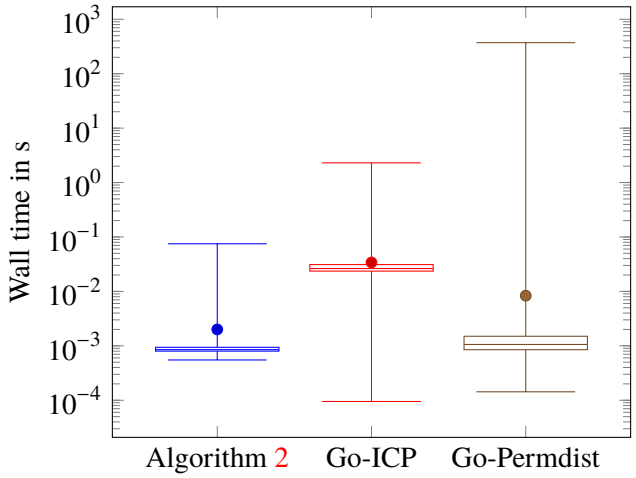
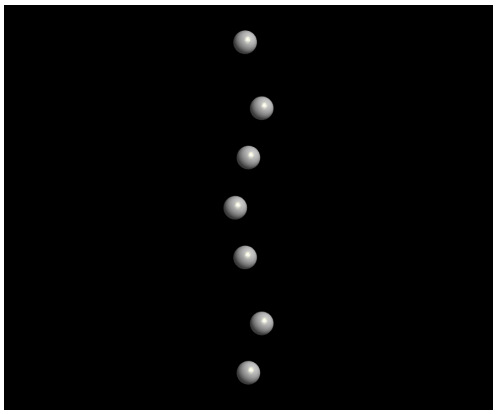


Figure 5.6: Distribution of the wall times in the silicon dataset.

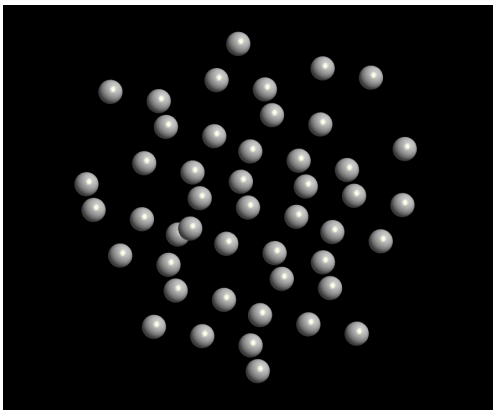
The first dataset under investigation is the silicon dataset. Figure 5.6 shows how the runtimes are distributed for the three different algorithms. The lines in this box plot show (starting from below) the 0% (minimum), 25%, 50% (median), 75% and 100% (maximum) percentile of the wall times. The average runtime is represented by the single dot. On average, algorithm 2 is about four times faster than Go-Permdist and ten times faster than Go-ICP. Interestingly, Go-ICP and Go-Permdist show a much larger variation in the runtimes than algorithm 2. Additional investigations showed that all three algorithms are slowest, when the neighbourhoods under investigation are very symmetric, for example when they are the neighbourhood of a particle in a perfect crystal. On the other hand, all algorithms performed best, when the neighbourhoods had low orders of symmetry, as shown in figure 5.7.

This behaviour can easily be explained. As Go-ICP and Go-Permdist search the space of all possible rotations using a branch-and-bound approach, large parts of the rotation space can immediately be excluded if the systems under investigation display almost no rotational symmetry. Algorithm 2 also benefits from low-symmetry systems as this limits the number of index combinations of length three that have to be tested.

Applying the three algorithms to the amorphous dataset yields results that are very similar to



(a) Low-symmetry neighbourhood that was involved in the fastest Go-Permdist calculation.



(b) High-symmetry neighbourhood that was involved in the slowest Go-Permdist calculation.

Figure 5.7: Extreme neighbourhoods in the silicon dataset.

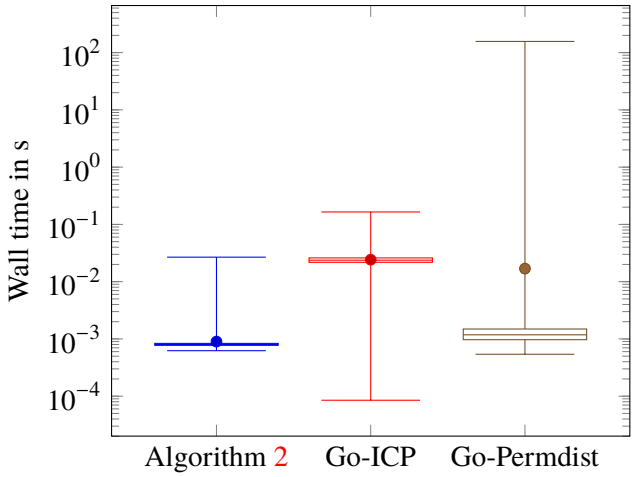


Figure 5.8: Distribution of the wall times in the amorphous dataset.

the results from the silicon dataset (see figure 5.8). Algorithm 2 is once again faster on average than Go-ICP and Go-Permdist. As before, Go-Permdist shows a huge variation in its runtimes, spanning several orders of magnitude.

The next dataset to be investigated is the diamond dataset. At this point only the neighbourhoods with cutoff radius $r = 6$ Å were used. As all neighbourhoods in it were calculated using a perfect diamond crystal, and thus highly symmetric, poor performance was expected. From

the results in figure 5.9, several observations can be made. To begin with, the new algorithm 2 is on average faster than both Go-ICP and Go-Permdist, this time by a larger margin. Compared to algorithm 2, Go-ICP needs about 50 times as much time on average, for Go-Permdist this figure rises to 500.

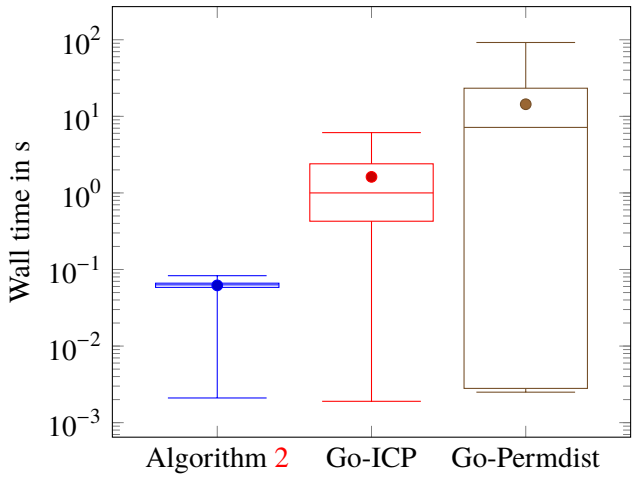


Figure 5.9: Distribution of the wall times in the diamond dataset ($r = 6$ Å).

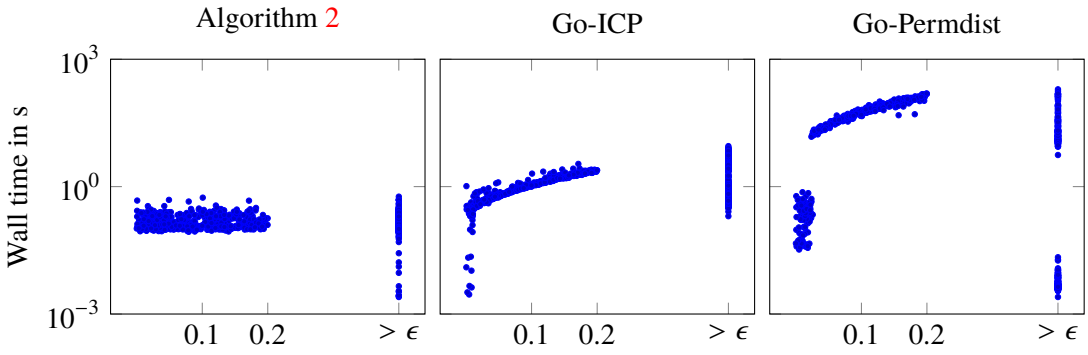


Figure 5.10: Relation between d^{full} and the runtimes for the diamond data set.

Once again, a large spread in the runtimes is visible. By looking at all neighbourhood pairs $(N_i, \tilde{N}_i)$ and studying the relation between $d^{full}(N_i, \tilde{N}_i)$ and the runtimes, some patterns emerge (see figure 5.10). Algorithm 2 needs approximately the same time for all pairs $(N_i, \tilde{N}_i)$ that fulfil $d^{full}(N_i, \tilde{N}_i) \leq \epsilon$. If $d^{full}(N_i, \tilde{N}_i) > \epsilon$, the algorithm may be much faster as a lot of possible permutations may be excluded. Go-ICP tends to be fastest if $d^{full}(N_i, \tilde{N}_i)$ is low and if

the optimal orthogonal transformation is close to the identity matrix. Go-Permdist seems to combine the properties of the two other algorithms.

It is worth pointing out that the results from the diamond dataset are extremely important for LKMC. Being able to compare neighbourhoods in a crystal in reasonable time is a basic requirement whenever LKMC is applied to a crystalline system, as in the ad-atom example in subsection 5.1.1. If a single comparison takes more than 10 seconds, as it does when using Go-Permdist, LKMC can probably never be used efficiently.

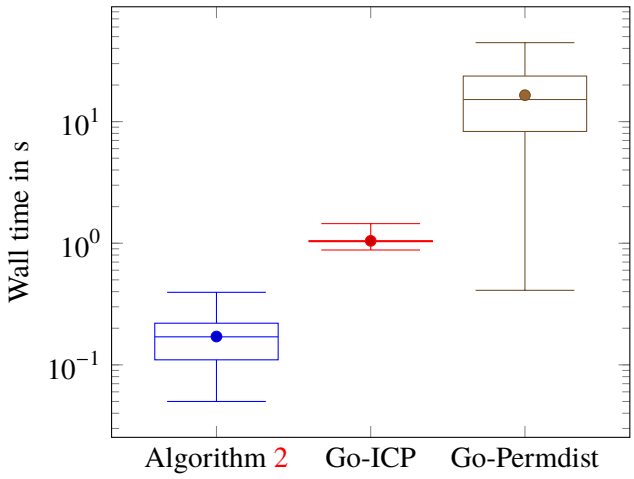


Figure 5.11: Distribution of the wall times in the sphere dataset ($n = 128$).

Finally, the neighbourhood pairs in the sphere dataset were compared, using only the neighbourhoods that contained 128 particles. As figure 5.11 shows, the results are qualitatively similar to the results from the diamond dataset. It is worth pointing out that although the neighbourhoods in the used sphere dataset (128 particles per neighbourhood) are smaller that the neighbourhoods in the diamond dataset (159 particles for $r = 6$ Å), the runtimes are higher on average for all the algorithms. This observation is explained in more detail in the subsequent scaling investigation.

5.3.3 Scaling with respect to n

In the previous section, the algorithms were compared only for fixed numbers of particles. How the algorithms scale with growing numbers of particles is investigated now. The first test was conducted on the diamond dataset. By choosing different cutoff radii r it is possible to generate neighbourhoods of different sizes. The worst-case runtimes for this dataset are shown in figure 5.12.

Somewhat unexpectedly, the runtimes of algorithm 2 seem to scale quadratically with the number of particles in the atomic neighbourhoods. This behaviour can be explained, however.

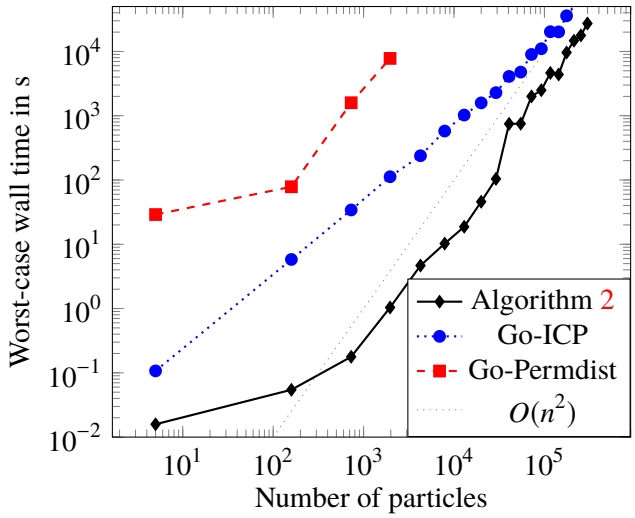


Figure 5.12: Worst-case wall times for the diamond example.

Due to the construction of the neighbourhoods, the particles in a neighbourhood lie on different shells around the central particle[3]. This shell-structure allows algorithm 2 to exclude a number of possible permutations as only particles with similar distance to the central particle may be matched. The Go-ICP algorithm seems to scale better than algorithm 2 but is still much slower. It should also be noted that the speed of Go-ICP (and Go-Permdist) depends not only on the number of particles but also on the tolerance δ in the stopping criterion. If this tolerance is lowered, the computational cost of Go-ICP increases considerably. Due to the high cost of the Go-Permdist algorithm, no results were calculated for neighbourhood pairs with more than 2000 particles. In figure 5.12 a relatively moderate increase in the worst-case wall time of algorithm 2 can be observed for low numbers of particles. This is most likely caused by the internals of the used QuantumATK software package (like license checks) which make comparisons of small neighbourhoods more expensive as they should be.

Next were the tests on the sphere dataset. In this dataset, all particles in the neighbourhoods (except for the central particle) lie on a sphere with some radius, thus only a single shell exists. Therefore, as expected, algorithm 2 shows the cubic scaling with the number of particles that was predicted in the complexity analysis (see figure 5.13). Here, Go-ICP shows similar scaling as algorithm 2 but is consistently slower by a factor of about 100.

[3] One could analyse this further by looking at the radial distribution function of an atom in a diamond crystal.

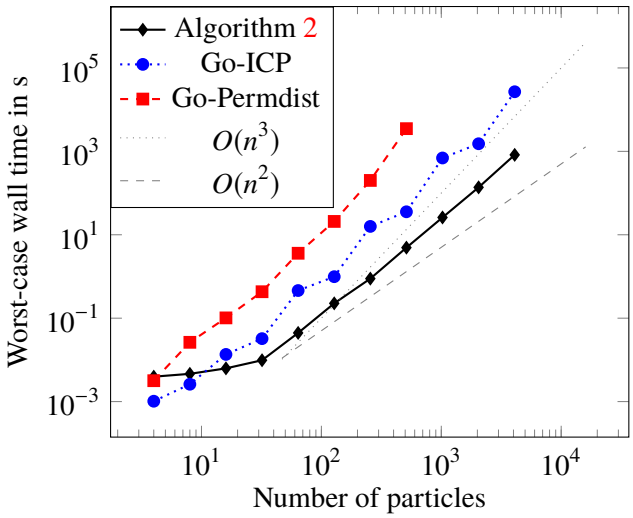


Figure 5.13: Worst-case wall times for the sphere example.

5.3.4 Scaling with respect to ϵ

Another important question is how the performance of the three algorithms varies with the tolerance ϵ. Algorithm 2 in its basic form supports only relatively small ϵ values, limited by the minimal particle distance. To test a wider range of ϵ-values, algorithm 2 was modified such that the calculation of $\bar{\pi}$ in theorem 6 degenerates to a brute-force permutation search if the optimal permutation is no longer given by a nearest-neighbour search.

Figure 5.14 shows the corresponding average wall times for the silicon example[4]. The cost of Go-ICP grows only slowly with increasing ϵ, while Go-Permdist becomes much more costly with larger ϵ. Algorithm 2 shows a moderate dependency on ϵ.

The results for the 128-particle sphere example in figure 5.15 are more interesting. The wall times for Go-ICP and Go-Permdist do not change with ϵ at all, while algorithm 2 becomes continuously more expensive – though for $\epsilon = 0.5\,\text{Å}$ it is still cheaper than Go-ICP and Go-Permdist. It was previously shown in figure 5.10 that Go-ICP and Go-Permdist can be fast if the actual distance between two neighbourhoods is very small. As the sphere dataset contains only neighbourhood pairs that are exactly equal (up to rotations and permutations), d^{full} is always zero here. To verify that the runtimes depend on the actual value d^{full}, the diamond dataset was also used, where noise was added to the neighbourhoods. As suspected, the cost of Go-ICP and Go-Permdist once again rises with the tolerance ϵ (see figure 5.16). For small ϵ, algorithm 2 behaves in a similar way as in the initial test on the silicon dataset. However, its

[4] As the tests that were performed here are relatively cheap, the obtained worst-case wall times were heavily affected by "background noise" caused by the operating system or the used hardware. Thus, the average wall times are used here that were much more robust.

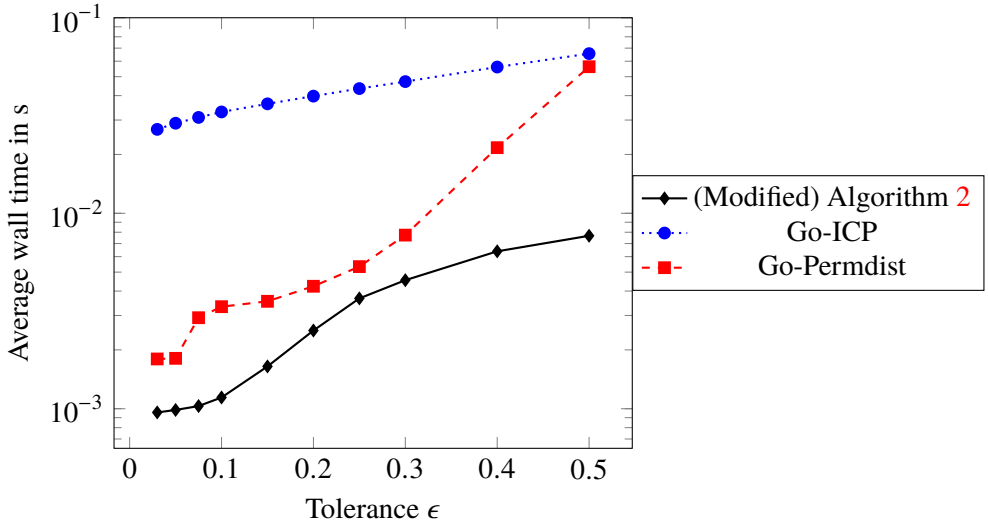


Figure 5.14: Average wall times for the silicon example.

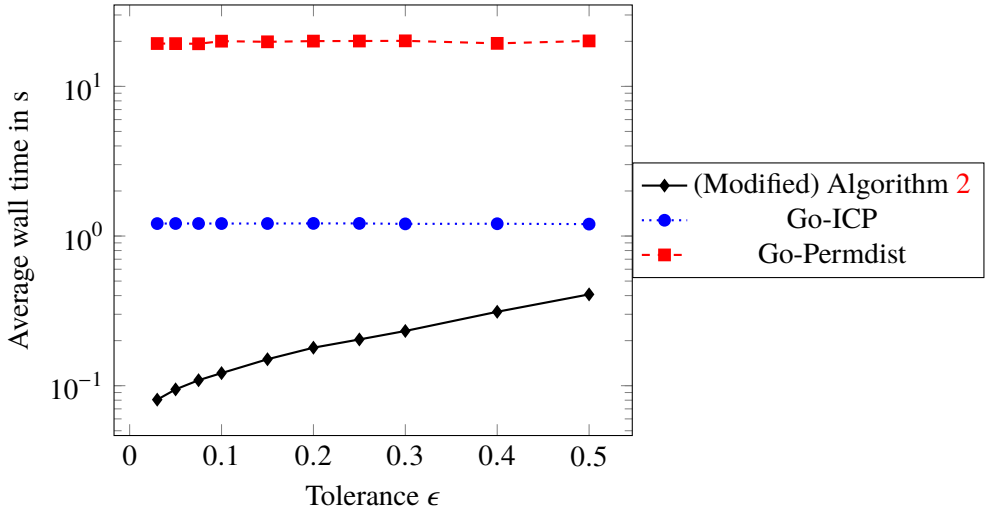


Figure 5.15: Average wall times for the 128-particle sphere example.

cost rises rapidly for $\epsilon > 0.4$ Å. The reason for this behaviour is that at this point ϵ becomes too large compared to the minimal particle distance μ that is used in algorithm 2. Therefore, the modified algorithm has to fall back to a (partial) brute-force search at this point which is much more computationally expensive.

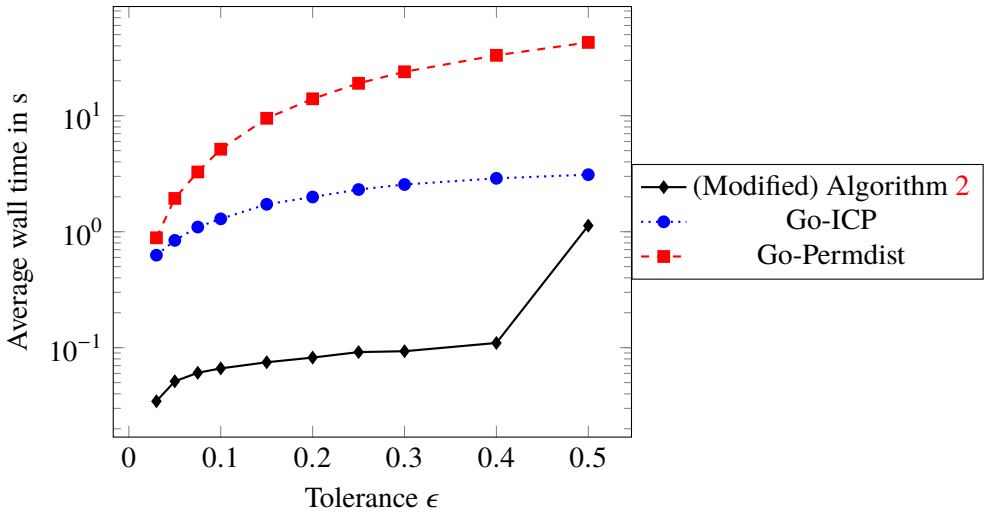


Figure 5.16: Average wall times for the diamond example ($r = 6$ Å).

5.3.5 Testing different invariances

One of the fundamental properties of the new comparison function is its invariance under both permutations and orthogonal transformations. It remains to be shown that both invariances are indeed necessary. To do this, the following test procedure is applied to all example problems in 5.1.

First, the AKMC algorithm in QuantumATK is used to calculate a number of KMC steps for each problem, thus generating a sequence of states (states that occurred previously are ignored). Then for each step, all the atomic neighbourhoods in the corresponding state are calculated and the number of new neighbourhoods is determined.

This procedure is used for different comparison functions. Besides the fully invariant RMSD d^{full}, the permutation invariant RMSD d^{perm} in equation (4.8) is used as well as d^{orth} (equation (4.7)) that is invariant under orthogonal transformations. The tolerance ϵ was set to 0.3 Å and the radius of the atomic neighbourhoods to 6 Å. Additionally, the nauty-based comparison approach that is used in k-ART is tested, using the same neighbourhood size as the RMSD-based measures and the radius 3 Å when building the adjacency graph.

Platinum ad-atom The first test uses the ad-atom example that was presented in 5.1.1. On a 55.5 Å × 55.5 Å [001] platinum surface, five ad-atoms were randomly placed and their evolution was simulated using the AKMC algorithm. Figure 5.17 shows the cumulative number of atomic neighbourhoods that were encountered during the AKMC simulation. Using d^{full} to compare atomic neighbourhoods yields similar results as using nauty. In the first KMC step, about 10 different neighbourhoods are found and almost no new neighbourhoods are encountered in

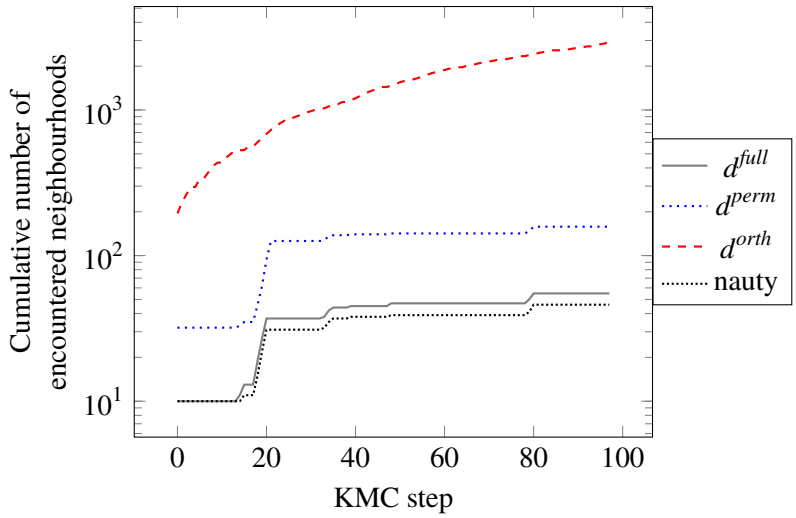


Figure 5.17: Ad-atom movement: number of new atomic environments in each KMC step.

the next 20 KMC steps. At this point, two ad-atoms have become so close that they start to interact and eventually form an ad-atom cluster. Therefore a number of new neighbourhoods are observed. As soon as the cluster formation has finished, the number of encountered neighbourhoods stays almost constant. Only a few new neighbourhoods are found as some of the ad-atoms may get close to the two-atom cluster and move away again.

The permutation invariant comparison function d^{perm} behaves qualitatively the same: as soon as the transition mechanisms of the isolated ad-atoms have been learned, almost no new neighbourhoods are encountered until the ad-atoms start to form clusters. Compared to d^{full}, the number of encountered neighbourhoods is almost ten times higher. Using only invariance under orthogonal transformations (d^{orth}) does not seem sufficient in this case. The number of encountered neighbourhoods grows in every KMC step, even if no new transition mechanisms occur.

Amorphous silicon Next in line is the amorphous silicon case from subsection 5.1.2 with five added hydrogen atoms. As shown in figure 5.18 there is almost no difference in the number of encountered neighbourhoods when comparing the three RMSD-based methods. This is not entirely unexpected, as the atomic neighbourhoods in the amorphous system do not show any obvious symmetries that can be exploited. Therefore, using d^{full} instead of a simple comparison function like d^{perm} or d^{orth} would most likely not be advantageous unless some parts of the amorphous system start to crystallise over a long KMC simulation. Using nauty, a somewhat lower number of atomic neighbourhoods is encountered. However, by analysing the neighbourhoods that nauty determined as equal it became apparent that the reason for

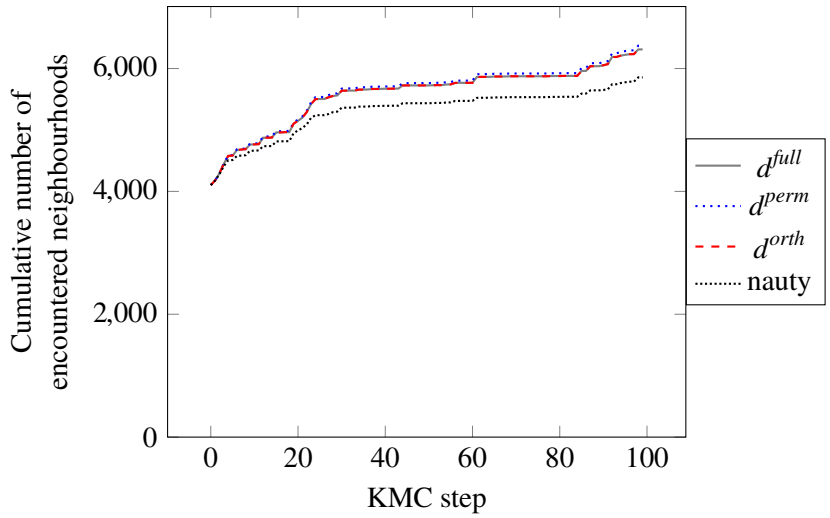


Figure 5.18: Amorphous silicon problem: number of new atomic environments in each KMC step.

this behaviour is that nauty produces false positives. One example is shown in figure 5.19 where the nauty-based approach does not capture the difference in the position of the hydrogen atom (marked blue). Some of the neighbourhoods that nauty determined to be similar showed differences in the coordinates of a single particle that were larger than 1 Å. This is far too high, given that only three proper transitions were encountered were a particle moves by more than 1 Å (see figure 5.20).

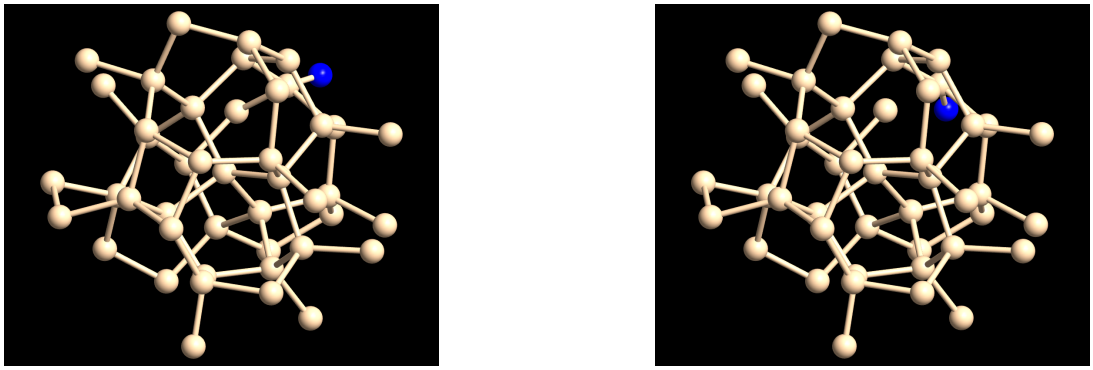


Figure 5.19: Two neighbourhoods that the nauty-based comparison determines to be almost equal.

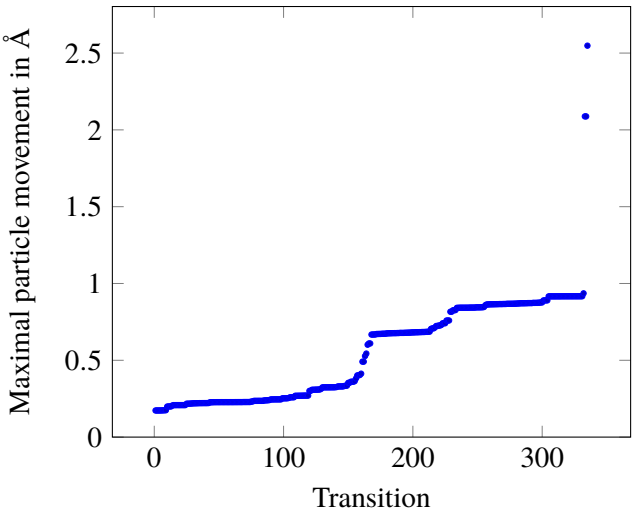


Figure 5.20: Amorphous silicon problem: distribution of maximal particle movement during a transition.

Nickel grain boundary The grain boundary problem in 5.1.3 combines the properties of crystalline and amorphous systems. While a large part of the initial system is perfectly crystalline, the interface between the two crystals has amorphous properties. This is reflected in the number of neighbourhoods that are encountered during a KMC simulation. Using the fully invariant RMSD offers some benefits compared to d^{perm} or d^{orth} though the difference is not as large as in the ad-atom case (see figure 5.21). As for the amorphous problem, nauty once again produces false positives, which also explains the somewhat lower number of encountered neighbourhoods, compared to d^{full}.

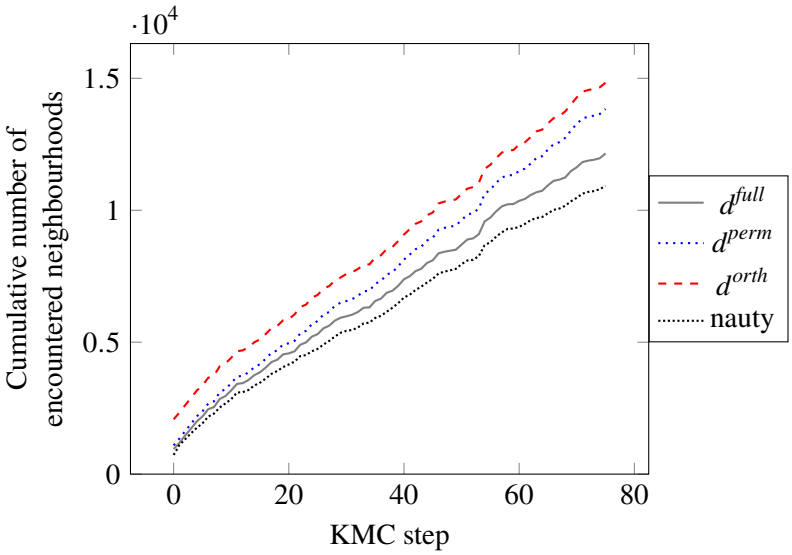


Figure 5.21: Grain boundary problem: number of new atomic environments in each KMC step.

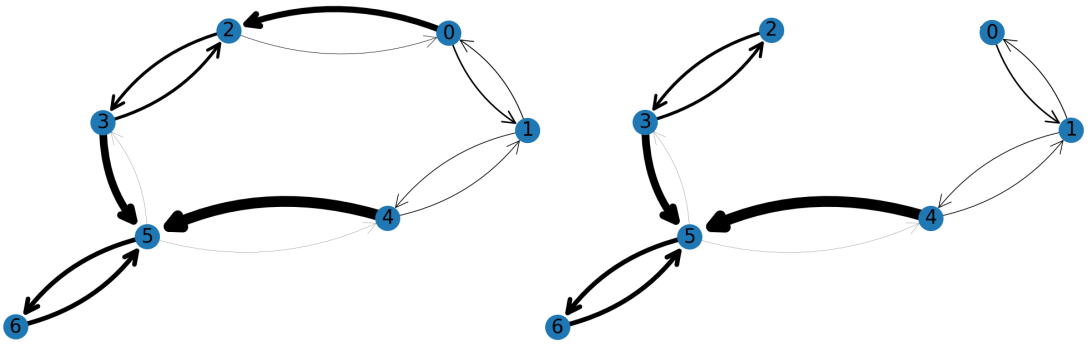
5.4 LKMC

Now it is time to test the LKMC algorithm itself. Two important properties are analysed: its performance as measured by the wall time required to calculate a given number of KMC steps, as well as its accuracy.

One important observation is that the transition searches in LKMC (and AKMC) are not deterministic as they are based on high-temperature Langevin trajectories. Additionally, the used stopping criterion (see subsection 4.7.2) estimates only the likelihood that all transitions were found. Thus, applying LKMC or AKMC to the same problem several times usually yields different results. Therefore, unless stated otherwise, median values for the wall times and for the accuracy are reported.

Measuring the accuracy is a complex task. For some simple toy problems, AKMC/LKMC may be able to recover the whole underlying Markov Jump Process. If more complex problems are used, this is often not the case as some transitions may be missed by the transition search procedure. Then the next question is which parts of the MJP are important? Many transitions with low transition rates may be unimportant. Others, however may be crucial as they can be the sole link to other parts of the MJP. The importance of transitions can also differ, depending on the length of a simulation. Consider the MJP in figure 5.22 where the thickness of the arrows corresponds to the transition rates. If an on-the-fly KMC simulation starts in state 0 and does not find the transition from 0 to 2, the short-term behaviour of the MJP is severely altered. In the long term, however, this does not matter much as the system will eventually switch between

the states 5 and 6 most of the time, irrespective of the original path from state 0 to 5.



(a) Complete MJP (b) MJP with one missing transition

Figure 5.22: Example: MJP with missing transitions.

No solution to this accuracy dilemma is given here. Instead, the accuracy definition from [41] is used. In this work, the total escape rates K_i are used as a proxy for the accuracy. These rates determine the average time that a system spends in a state of the underlying MJP and are therefore important for the simulation of the dynamics. To investigate the accuracy, a short AKMC simulation was run for each test problem in order to calculate a sequence of states $\sigma_1, \ldots, \sigma_m$ (duplicates were dropped). An excessive number of global transition searches were then applied to each such state σ_i in order to find as many transitions from this state as possible. These transitions were then used to calculate the reference total escape rate $K_i^{reference}$ from state σ_i.

Afterwards, the transition search functionality in AKMC and LKMC was applied to each state, yielding the wall times, the approximated total escape rates K_i^{AKMC} and K_i^{LKMC}, and the corresponding relative error

$$\frac{|K_i^{AKMC/LKMC} - K_i^{reference}|}{K_i^{reference}}.$$

The total transition rates $K_i^{AKMC/LKMC}$ were calculated multiple times in order to calculate statistical averages like the median relative error.

5.4.1 Platinum ad-atom

As in the previous experiments, a 55.5 Å × 55.5 Å [001] platinum surface with five randomly placed ad-atoms was simulated at 300 K. The full list of used parameters can be found in the appendix (A.1). Accuracy-wise, neither AKMC nor LKMC had problems with this problem. No matter which algorithm (or cutoff radius) is chosen, the median error in the escape rate is always less than 0.1%.

One interesting observation points to problems with the used HTST rate theory and/or the used potential. Usually, the energy barrier of a particle hopping on the platinum surface is about 0.93 eV but there seems to be another (much longer and probably unphysical) reaction path that has a lower energy barrier of 0.82 eV. This longer path is sometimes found by the AKMC algorithm and also by the EON software [100, 101] which is a related on-the-fly KMC implementation.

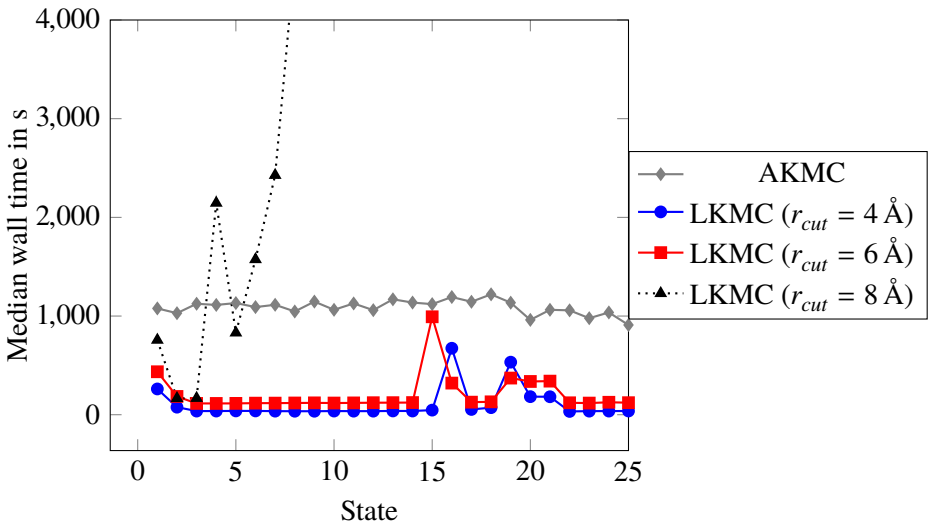


Figure 5.23: Ad-atom problem: median wall times.

The wall time plot in figure 5.23 is also interesting. As expected, AKMC needs about the same time for every KMC step as it has to recalculate the transitions in every step. LKMC with low cutoff radii (4 Å and 6 Å) also works as expected. In the first KMC step, a lot of computational time is needed to determine all possible local transition mechanisms. Afterwards, these mechanisms are simply recycled, leading to much lower wall times per step. At some point (around step 15), some ad-atoms begin to approach each other, causing new local structures which requires LKMC to run new local transition searches. As soon as these new transition mechanisms have been learned, the required wall time per step returns to its former lower level.

A big outlier is the LKMC algorithm with a neighbourhood cutoff of 8 Å. While it produces the same results in term of accuracy as the other variants, it is much slower and may need up to 12,000 seconds per KMC step (not visible in the plots). This big drop in performance can be explained by looking at how local transition searches are set up. In section 4.1 it was shown that under certain circumstances, local transition searches can approximate global transitions quite accurately. However, local transition searches may also find transitions that do not occur in the global case. In this example, problems occur when one ad-atom is close to the centre of an atomic neighbourhood while another is close to its edge. The ad-atom close to the edge

of the atomic neighbourhood may then behave in an unphysical way as the particles outside the atomic neighbourhood are constrained. Therefore, LKMC calculates a vast number of unnecessary local transitions and also unsuccessfully tries to apply them to the global system, causing a huge waste of computational time. This behaviour could be avoided by checking if a local transition also works in the global context. This feature has not yet been implemented, as this would require larger changes to the used transition search mechanism in QuantumATK.

The ad-atom problem is also a prime example to show the advantages of (general) on-the-fly KMC algorithms. The first 25 KMC steps of the reference calculation alone, cover a time interval of about 35 µs. Reaching a similar timescale by discretising the Langevin equations would require 3.5×10^{10} timesteps (using a timestep length of 1 fs).

5.4.2 Hydrogen diffusion in amorphous silicon

The same kind of analysis was also applied to the hydrogen diffusion problem. Here, the movement of five hydrogen atoms in the 4096-particle amorphous silicon system was simulated at 300 K (other parameters in appendix A.2). In this example, the first 25 KMC timesteps cover about 0.6 ns, though the time that is covered per KMC step increases as the system settles. Looking at the wall times in figure 5.24, the results look sobering. There is no clear speed advantage, neither for LKMC nor for AKMC. On a closer look, however, the three tested LKMC variants show a downward trend in the wall time that is required for finding the transitions. This is an effect of the increasing number of stored local transition data in the LKMC database, which lowers the number of necessary transition searches.

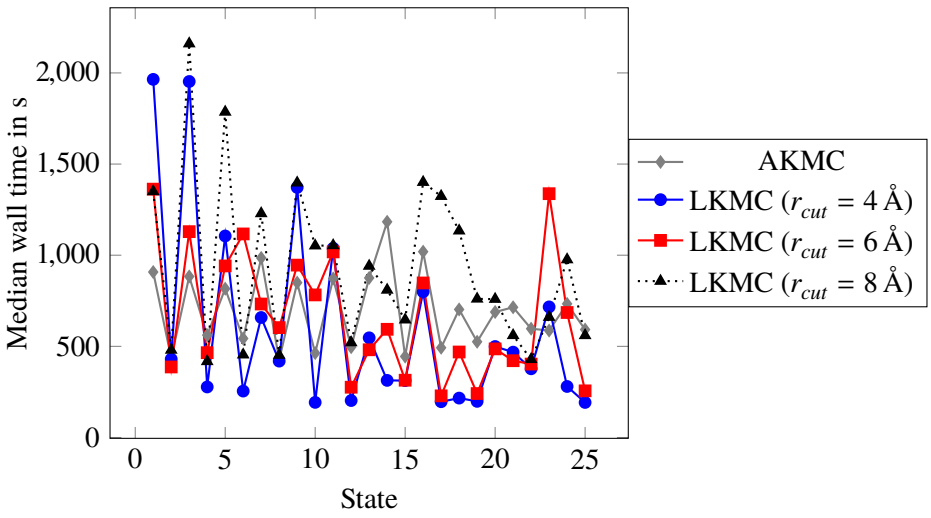


Figure 5.24: Amorphous silicon problem: median wall times.

The accuracy results look promising: in the first few steps (see figure 5.25), the error in the

escape rates is approximately the same for all compared methods. This was to be expected as LKMC essentially degenerates to AKMC if no local transition information is available. After a few steps, however, the median escape rate error of the LKMC algorithm becomes much lower than the corresponding value for the AKMC algorithm, which may miss up to 40% of the total transition rate. As investigated earlier (see figure 5.4), cutoff values of at least 6 Å are required to accurately capture most of the transitions. For r_{cut} = 8 Å, two states (20 and 24) show abnormally high relative errors. In both states, one transition with a high transition rate exists that is not always found by the LKMC algorithm with cutoff 8 Å. As median values for the relative error are reported, the error is large for these two states[5]. The reason for this behaviour could not be identified with certainty. Most likely, the used saddle search algorithm in QuantumATK is very sensitive with respect to its initial guess. As the initial guess is extracted from a (non-deterministic) high-temperature Langevin trajectory, the corresponding saddle searches may succeed or fail in a seemingly random fashion.

Compared to the AKMC results, the errors that are obtained with the LKMC algorithm show a slight downwards trend with respect to the KMC steps. This could be an indication that the approach to learn local transitions actually shows its benefits. It would definitely be interesting to investigate the behaviour of these curves for an even larger number of KMC steps. Due to the extremely high cost of calculating reference values for the total escape rates, this would however need a prohibitive amount of computational power.

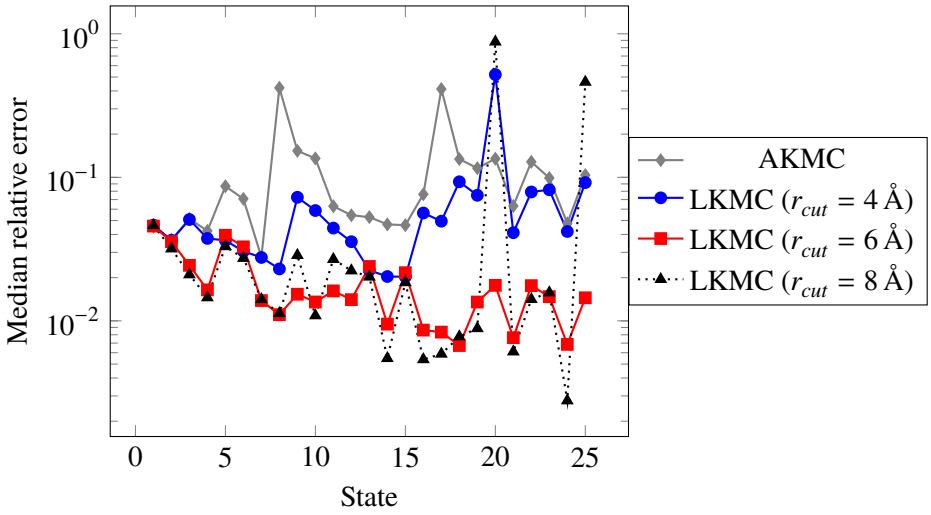


Figure 5.25: Amorphous silicon problem: relative errors in the total escape rates.

[5] The important transition in state 20 is found in about 40% of the runs.

5.4.3 Nickel grain boundary

For the grain boundary problem, a temperature of 500 K was used as well as the parameters in table A.3. The previous analysis of the localisability in figure 5.5 predicts that relatively large cutoff radii are needed to capture all transitions.

This prediction seems to be true, as shown in figure 5.26. If small cutoff values (< 8 Å) are used, the LKMC algorithm misses many transitions, leading to relative errors in the total escape rates that can be close to 100%. If larger cutoffs are used, LKMC behaves similar to AKMC, yielding almost the same escape rate errors.

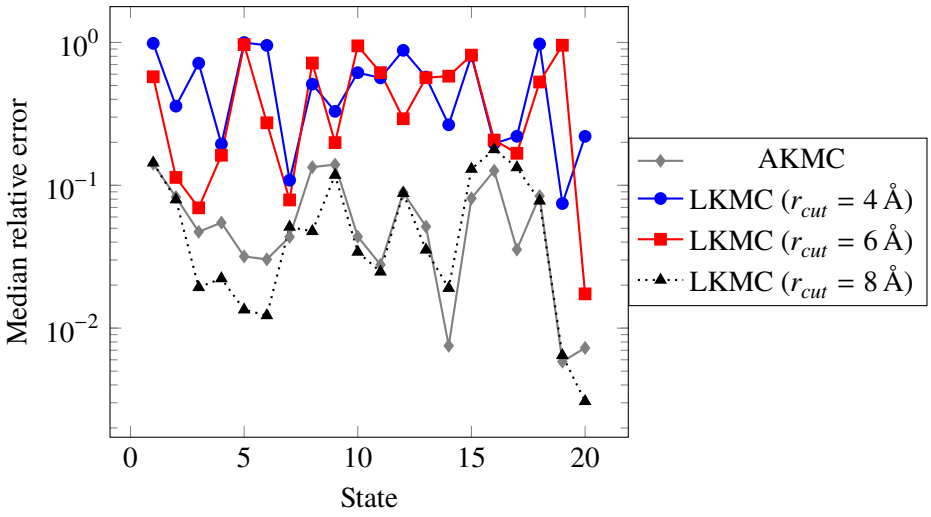


Figure 5.26: Grain boundary problem: relative errors in the total escape rates.

Speed-wise, LKMC is only distinctly faster than AKMC when using a low cutoff of 4 Å (see figure 5.27). As its accuracy is abysmal in this case, this is unfortunately of no use. Using larger cutoffs, LKMC is not faster than AKMC but much slower. This performance penalty is caused by a number of factors: as in the ad-atom case, the local transition searches find a number of local transitions that cannot be applied to the global system. Furthermore, the cost of neighbourhood comparisons rises with the cutoff radii, as well as the number of distinct neighbourhoods.

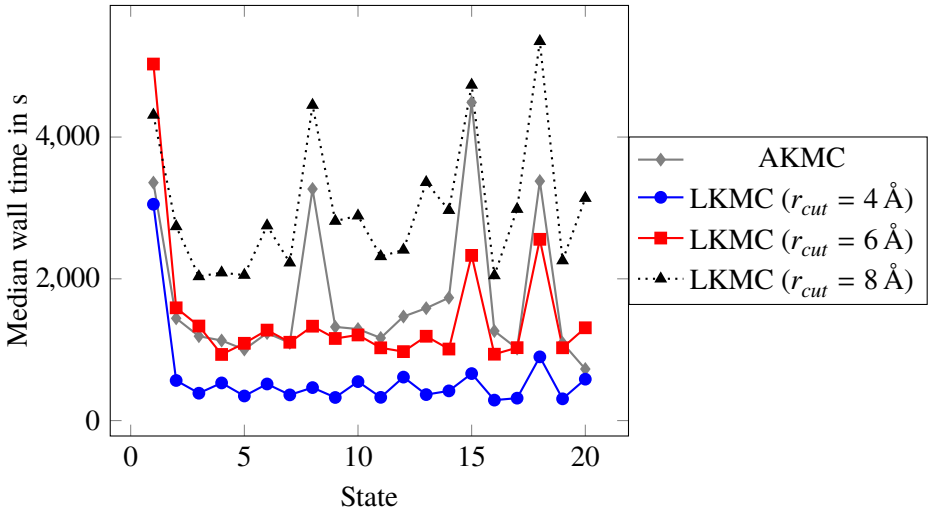


Figure 5.27: Grain boundary problem: median wall times.

CHAPTER 6

Conclusions

The main goal of this thesis was to modify the k-ART algorithm such that it can be applied to amorphous systems. The key component to achieve this is the new algorithm 2 that compares particle systems using an invariant RMSD. Unlike other algorithms that calculate the invariant RMSD exactly, it is fast enough to compare a large number of systems in reasonable time. This makes it possible to build databases that can be used to query for particle systems and corresponding data, even if they have been affected by orthogonal transformations and index permutations. Here, this was used to store atomic neighbourhoods and corresponding local transition mechanisms but several other applications are possible. Instead of building a local transition database, results from expensive QM calculations could be stored in a database and reused, when necessary. To my knowledge, no other database has been proposed, that can query for molecules or atomic neighbourhoods in such an invariant way, calculate the corresponding globally optimal transformations, and that does not suffer from extremely long query times.

The invariant comparison procedure could also be used for the analysis of local structures in molecules and crystals in order to identify similar regions in a particle system or simply to classify different local structures. Another possible application is machine learning. Many machine learning algorithms that try to learn properties of particle systems have to determine the degree of similarity between two particle systems (see for example [67]). Using the invariant RMSD in such a setting would be interesting, though it is not clear whether algorithm 2 is fast enough for such applications.

While algorithm 2 was faster than the reference algorithms Go-ICP and Go-Permdist in the evaluation, it has a much more limited scope than these other two algorithms. By design, it can only test for similarity when the comparison tolerance ϵ is small and cannot be applied if this is not the case. Still, this seems to be sufficient to use it in database applications.

Using the invariant RMSD for comparison in the LKMC algorithm offers advantages over both k-ART and classical on-the-fly KMC algorithms. Replacing the graph based comparison in k-ART by this new approach does away with two of k-ART's major problems: it eliminates

the possibility of false positives when comparing (amorphous) neighbourhoods and makes it possible to map local transitions from one neighbourhood to another without resorting to heuristic approaches. Compared to the reference on-the-fly KMC algorithm, it can provide better accuracy and/or performance by recycling known local transitions. If the simulated system is amorphous and stays amorphous during the whole simulation, using the fully invariant RMSD does not seem to be necessary. In this case it may be sufficient to drop one of the invariances and work with a simpler version of it. However, if some parts of an amorphous system start to crystallise, dropping the invariance may cause a substantial performance penalty, as the local transitions in the crystalline parts may not be reused.

As expected, the usefulness of LKMC (as well of k-ART) rises or falls with the localisability of the underlying problem. This was very much visible in the grain boundary problem where badly localised transitions caused LKMC to produce inaccurate results. However, this does not mean that LKMC is in general unsuited for such grain boundary problems. While all used problems were not simple toy problems, they were also not truly realistic. Setting up a truly realistic simulation is a difficult problem on its own, as it requires finding (or creating) a suitable potential energy function as well as an initial condition for the particle positions. Thus, the used grain boundary problem is not representative for all grain boundary simulations.

Some unsolved problems remain. It was always assumed here, that the used HTST rate theory is sufficiently accurate. In the ad-atom example problem, however, more than one reaction path between two states was found which violates the assumptions in HTST and therefore puts its validity into question. Furthermore, it has not been established that HTST can be applied to amorphous systems at all.

One major issue that LKMC cannot solve in its current form, is that the time that is covered by a single KMC step usually declines when the particle system under investigation becomes more complex. This was an issue when I tried to simulate the deposition of silicon atoms on a silicon surface as stated in the introduction. As more and more particles were deposited onto the surface, the number of possible transitions grew and the average time per KMC step declined. While LKMC was faster than the reference on-the-fly KMC algorithm in this case, it was not any faster than a plain Langevin simulation, as soon as a certain number of particles had been deposited onto the surface. Thus, it would be interesting to see whether LKMC could be extended to solve this problem. One idea would be to use the LKMC database to find neighbourhoods in a particle system that do not allow local transitions. If these “inert” neighbourhoods separate the system into subregions, decoupled LKMC simulations could be run in these subregions.

APPENDIX A

Appendix

A.1 Fixing Go-Permdist

The Go-Permdist algorithm in [69] applies a branch-and-bound approach to the space of all rotations. In this algorithm, rotations are represented by a vector $r \in [-\pi, \pi]^3$ where $\|r\|_2$ denotes the rotation angle and $r/\|r\|_2$ the rotation axis.

The Go-Permdist algorithm [69] tries to determine d^{full} by decomposing the space of all rotations into *rotation cubes* $C(v, \theta_B) = \{r \in [-\pi, \pi]^3 : \|r - v\|_\infty \leq \theta_B\}$. Each vector r in a rotation cube represents a rotation of angle $\|r\|_2$ around the rotation axis $r/\|r\|_2$. For simplicity $r(x)$ is used to denote a vector x that has been rotated by r. In order to work, the Go-Permdist algorithm needs a lower bound $LB(v, \theta_B)$ that depends only on v and θ_B such that

$$LB(v, \theta_B) \leq \cos(\angle(v(x), r(x))), \quad \forall r \in C(v, \theta_B), x \in \mathbb{R}^3.$$

In equation 19 of the cited paper such a lower bound is derived that unfortunately does not hold true in all cases. As a simple counter example $v = (-0.25\pi, 0.25\pi, 0.25\pi)$, $r = (-0.48\pi, 0.02\pi, 0)$, $x = (0.12\pi, -0.14\pi, 0)$ and $\theta_B = 0.5\pi$ can be chosen. Fortunately, the problem seems to be just a missing factor $\sqrt{3}$. By adding it to equation 19 in the original paper the following corrected lower bound was obtained:

$$LB(v, \theta_B) = \cos\left(\min\left[\pi, \sqrt{3}\frac{\theta_B}{2}\right]\right).$$

A.2 Simulation parameters

This section lists the parameters that were used during the evaluation in chapter 5.

AKMC and LKMC	
CPU-Cores	32 (Intel Xeon Gold 6130)
KMC temperature	300 K
Saddle search temperature	2 000 K
Confidence	0.99
Harmonic prefactor	Fixed value of 1×10^{13} Hz
LKMC	
Interaction radius $r_{interact}$	6 Å

Table A.1: AKMC/LKMC parameters for the ad-atom simulation in subsection 5.4.1.

AKMC and LKMC	
CPU-Cores	32 (Intel Xeon Gold 6130)
KMC temperature	300 K
Saddle search temperature	1 200 K
Confidence	0.99
Harmonic prefactor	Fixed value of 1×10^{13} Hz
LKMC	
Interaction radius $r_{interact}$	6 Å

Table A.2: AKMC/LKMC parameters for the hydrogen diffusion in amorphous silicon in subsection 5.4.2.

AKMC and LKMC	
CPU-Cores	32 (Intel Xeon Gold 6130)
KMC temperature	500 K
Saddle search temperature	1 500 K
Confidence	0.99
Harmonic prefactor	Fixed value of 1×10^{13} Hz
LKMC	
Interaction radius $r_{interact}$	6 Å

Table A.3: AKMC/LKMC parameters for the grain boundary simulation in subsection 5.4.3.

Bibliography

[1] D. Harries and G. Broomfield, *Hydrogen Embrittlement of Steel Pressure Vessels in Pressurised Water Reactor Systems*. AERE-R, UK Atomic Energy Authority Research Group, 1962.

[2] T. Allen, J. Busby, M. Meyer, and D. Petti, "Materials challenges for nuclear systems," *Materials Today*, vol. 13, no. 12, pp. 14 – 23, 2010.

[3] L. R. Bailey, G. Proudfoot, B. Mackenzie, N. Andersen, A. Karlsson, and A. Ulyashin, "High rate amorphous and crystalline silicon formation by pulsed DC magnetron sputtering deposition for photovoltaics," *physica status solidi (a)*, vol. 212, no. 1, pp. 42–46, 2015.

[4] A. N. Naganathan and V. Muñoz, "Scaling of folding times with protein size," *Journal of the American Chemical Society*, vol. 127, no. 2, pp. 480–481, 2005. PMID: 15643845.

[5] H. J. C. Berendsen, *Simulating the Physical World: Hierarchical Modeling from Quantum Mechanics to Fluid Dynamics*. New York, NY, USA: Cambridge University Press, 2007.

[6] E. Tadmor and R. Miller, *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*. Cambridge University Press, 2011.

[7] M. Griebel, S. Knapek, and G. Zumbusch, *Numerical Simulation in Molecular Dynamics: Numerics, Algorithms, Parallelization, Applications*. Springer Publishing Company, Incorporated, 1st ed., 2007.

[8] M. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation*. Oxford Graduate Texts, OUP Oxford, 2010.

[9] A. Leach, *Molecular Modelling: Principles and Applications*. Prentice Hall, 2001.

[10] N. Tchipev, S. Seckler, M. Heinen, J. Vrabec, F. Gratl, M. Horsch, M. Bernreuther, C. W. Glass, C. Niethammer, N. Hammer, B. Krischok, M. Resch, D. Kranzlmüller, H. Hasse, H.-J. Bungartz, and P. Neumann, "Twetris: Twenty trillion-atom simulation," *The International Journal of High Performance Computing Applications*, 2019.

[11] P. V. Coveney and S. Wan, "On the calculation of equilibrium thermodynamic properties from molecular dynamics," *Phys. Chem. Chem. Phys.*, vol. 18, pp. 30236–30240, 2016.

[12] R. Palmer, "Broken ergodicity," *Advances in Physics*, vol. 31, no. 6, pp. 669–735, 1982.

[13] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, "Molecular dynamics with coupling to an external bath," *The Journal of Chemical Physics*, vol. 81, no. 8, pp. 3684–3690, 1984.

[14] W. G. Hoover and B. L. Holian, "Kinetic moments method for the canonical ensemble distribution," *Physics Letters A*, vol. 211, no. 5, pp. 253 – 257, 1996.

[15] T. Lelièvre and G. Stoltz, "Partial differential equations and stochastic methods in molecular dynamics," *Acta Numerica*, vol. 25, p. 681–880, 2016.

[16] F. Conrad and M. Grothaus, "Construction, ergodicity and rate of convergence of N-particle Langevin dynamics with singular potentials," *Journal of Evolution Equations*, vol. 10, pp. 623–662, Aug 2010.

[17] D. P. Herzog and J. C. Mattingly, "Ergodicity and Lyapunov functions for Langevin dynamics with singular potentials," *arXiv e-prints*, Nov 2017.

[18] H. A. Posch, W. G. Hoover, and F. J. Vesely, "Canonical dynamics of the Nosé oscillator: Stability, order, and chaos," *Phys. Rev. A*, vol. 33, pp. 4253–4265, Jun 1986.

[19] P. K. Patra and B. Bhattacharya, "Nonergodicity of the Nose-Hoover chain thermostat in computationally achievable time," vol. 90, p. 043304, Oct. 2014.

[20] E. Braun, S. M. Moosavi, and B. Smit, "Anomalous effects of velocity rescaling algorithms: The flying ice cube effect revisited," *Journal of Chemical Theory and Computation*, vol. 14, no. 10, pp. 5262–5272, 2018. PMID: 30075070.

[21] N. Goga, A. J. Rzepiela, A. H. de Vries, S. J. Marrink, and H. J. C. Berendsen, "Efficient Algorithms for Langevin and DPD Dynamics," *Journal of Chemical Theory and Computation*, vol. 8, no. 10, pp. 3637–3649, 2012. PMID: 26593009.

[22] R. Mathias, S. Gabriel, and L. Tony, *Free Energy Computations: A Mathematical Perspective*. World Scientific Publishing Company, 2010.

[23] G. Di Gesu, T. Lelievre, D. Le Peutrec, and B. Nectoux, "Jump Markov models and transition state theory: the quasi-stationary distribution approach," *Faraday Discuss.*, vol. 195, pp. 469–495, 2016.

[24] M. Evstigneev and P. Reimann, "Langevin equation for a system nonlinearly coupled to a heat bath," *Phys. Rev. B*, vol. 82, p. 224303, Dec 2010.

[25] R. Kupferman and A. Stuart, "Fitting SDE models to nonlinear Kac–Zwanzig heat bath models," *Physica D: Nonlinear Phenomena*, vol. 199, no. 3, pp. 279 – 316, 2004.

[26] H. Kramers, "Brownian motion in a field of force and the diffusion model of chemical reactions," *Physica*, vol. 7, no. 4, pp. 284 – 304, 1940.

[27] A. Bovier and F. den Hollander, *Metastability: A Potential-Theoretic Approach*. Grundlehren der mathematischen Wissenschaften, Springer International Publishing, 2016.

[28] M. Peleg, M. D. Normand, and M. G. Corradini, "The Arrhenius equation revisited," *Critical Reviews in Food Science and Nutrition*, vol. 52, no. 9, pp. 830–851, 2012. PMID: 22698273.

[29] M. Menzinger and R. Wolfgang, "The meaning and use of the Arrhenius activation energy," *Angewandte Chemie International Edition in English*, vol. 8, no. 6, pp. 438–444, 1969.

[30] P. Collet, S. Martínez, and J. San Martin, *Quasi-stationary distributions. Markov chains, diffusions and dynamical systems*. Probability and its applications, Springer, 2013.

[31] H. Eyring, "The activated complex in chemical reactions," *The Journal of Chemical Physics*, vol. 3, no. 2, pp. 107–115, 1935.

[32] E. Vanden-Eijnden and F. A. Tal, "Transition state theory: Variational formulation, dynamical corrections, and error estimates," *The Journal of Chemical Physics*, vol. 123, no. 18, p. 184103, 2005.

[33] P. Pechukas and E. Pollak, "Classical transition state theory is exact if the transition state is unique," *The Journal of Chemical Physics*, vol. 71, no. 5, pp. 2062–2068, 1979.

[34] G. H. Vineyard, "Frequency factors and isotope effects in solid state rate processes," *Journal of Physics and Chemistry of Solids*, vol. 3, no. 1, pp. 121 – 127, 1957.

[35] C. T. Campbell, L. Árnadóttir, and J. R. V. Sellers, "Kinetic prefactors of reactions on solid surfaces," *Zeitschrift für Physikalische Chemie*, vol. 227, no. 11, pp. 1435 – 1454, 2013.

[36] M. R. Sørensen and A. F. Voter, "Temperature-accelerated dynamics for simulation of infrequent events," *The Journal of Chemical Physics*, vol. 112, no. 21, pp. 9599–9606, 2000.

[37] E. Vanden-Eijnden, *Transition Path Theory*, pp. 453–493. Springer Berlin Heidelberg, 2006.

[38] C. Schütte, W. Huisinga, and P. Deuflhard, "Transfer operator approach to conformational dynamics in biomolecular systems," in *Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems*, (Berlin, Heidelberg), pp. 191–223, Springer Berlin Heidelberg, 2001.

[39] A. F. Voter, "Introduction to the kinetic monte carlo method," in *Radiation Effects in Solids*, (Dordrecht), pp. 1–23, Springer Netherlands, 2007.

[40] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[41] S. T. Chill and G. Henkelman, "Molecular dynamics saddle search adaptive kinetic Monte Carlo," *The Journal of Chemical Physics*, vol. 140, no. 21, p. 214110, 2014.

[42] N. Mousseau, L. Béland, P. Brommer, J.-F. Joly, F. El Mellouhi, E. Machado-Charry, M.-C. Marinica, and P. Pochet, "The activation-relaxation technique: Art nouveau and kinetic art," vol. 2012, pp. 1–14, 04 2012.

[43] H. Xu, Y. N. Osetsky, and R. E. Stoller, "Self-evolving atomistic kinetic Monte Carlo: fundamentals and applications," *Journal of Physics: Condensed Matter*, vol. 24, no. 37, p. 375402, 2012.

[44] A. F. Voter, "Hyperdynamics: Accelerated molecular dynamics of infrequent events," *Phys. Rev. Lett.*, vol. 78, pp. 3908–3911, May 1997.

[45] A. F. Voter, "Parallel replica method for dynamics of infrequent events," *Phys. Rev. B*, vol. 57, pp. R13985–R13988, Jun 1998.

[46] A. Pedersen, G. Henkelman, J. Schiøtz, and H. Jónsson, "Long time scale simulation of a grain boundary in copper," *New Journal of Physics*, vol. 11, no. 7, p. 073034, 2009.

[47] G. Henkelman and H. Jónsson, "Multiple time scale simulations of metal crystal growth reveal the importance of multiatom surface processes," *Phys. Rev. Lett.*, vol. 90, p. 116101, Mar 2003.

[48] D. S. D. Gunn, N. L. Allan, and J. A. Purton, "Adaptive kinetic Monte Carlo simulation of solid oxide fuel cell components," *J. Mater. Chem. A*, vol. 2, pp. 13407–13414, 2014.

[49] H. Xu, Y. N. Osetsky, and R. E. Stoller, "Simulating complex atomistic processes: On-the-fly kinetic Monte Carlo scheme with selective active volumes," *Phys. Rev. B*, vol. 84, p. 132103, Oct 2011.

[50] F. El-Mellouhi, N. Mousseau, and L. J. Lewis, "Kinetic activation-relaxation technique: An off-lattice self-learning kinetic Monte Carlo algorithm," *Phys. Rev. B*, vol. 78, p. 153202, Oct 2008.

[51] D. Konwar, V. J. Bhute, and A. Chatterjee, "An off-lattice, self-learning kinetic Monte Carlo method using local environments," *The Journal of Chemical Physics*, vol. 135, no. 17, p. 174103, 2011.

[52] Y. Shim, N. B. Callahan, and J. G. Amar, "Localized saddle-point search and application to temperature-accelerated dynamics," *The Journal of Chemical Physics*, vol. 138, no. 9, p. 094101, 2013.

[53] A. V. Fiacco, "Sensitivity analysis for nonlinear programming using penalty methods," *Mathematical Programming*, vol. 10, pp. 287–311, Dec 1976.

[54] P. Nylen, T.-Y. Tam, and F. Uhlig, "On the eigenvalues of principal submatrices of normal, hermitian and symmetric matrices," *Linear and Multilinear Algebra*, vol. 36, no. 1, pp. 69–78, 1993.

[55] L. K. Béland, P. Brommer, F. El-Mellouhi, J.-F. Joly, and N. Mousseau, "Kinetic activation-relaxation technique," vol. 84, p. 046704, Oct. 2011.

[56] L. Xu and G. Henkelman, "Adaptive kinetic Monte Carlo for first-principles accelerated dynamics," *The Journal of Chemical Physics*, vol. 129, no. 11, p. 114104, 2008.

[57] S. Y. Kim, D. Perez, and A. F. Voter, "Local hyperdynamics," *The Journal of Chemical Physics*, vol. 139, no. 14, p. 144110, 2013.

[58] M. A. Collins and D. F. Parsons, "Implications of rotation–inversion–permutation invariance for analytic molecular potential energy surfaces," *The Journal of Chemical Physics*, vol. 99, no. 9, pp. 6756–6772, 1993.

[59] "nauty software package." www.pallini.di.uniroma1.it/.

[60] B. D. McKay and A. Piperno, "Practical graph isomorphism, II," *Journal of Symbolic Computation*, vol. 60, no. 0, pp. 94 – 112, 2014.

[61] M. Trochet, A. Sauvé-Lacoursière, and N. Mousseau, "Algorithmic developments of the kinetic activation-relaxation technique: Accessing long-time kinetics of larger and more complex systems," *The Journal of Chemical Physics*, vol. 147, no. 15, p. 152712, 2017.

[62] K. C. Alexander and C. A. Schuh, "Towards the reliable calculation of residence time for off-lattice kinetic Monte Carlo simulations," *Modelling and Simulation in Materials Science and Engineering*, vol. 24, p. 065014, aug 2016.

[63] J.-F. Joly, L. K. Béland, P. Brommer, and N. Mousseau, "Contribution of vacancies to relaxation in amorphous materials: A kinetic activation-relaxation technique study," *Phys. Rev. B*, vol. 87, p. 144204, Apr 2013.

[64] M. Trochet, N. Mousseau, L. K. Béland, and G. Henkelman, *Off-Lattice Kinetic Monte Carlo Methods*, pp. 1–29. Cham: Springer International Publishing, 2019.

[65] M. C. Shaughnessy and R. E. Jones, "Efficient use of an adapting database of ab initio calculations to generate accurate Newtonian dynamics," *Journal of Chemical Theory and Computation*, vol. 12, no. 2, pp. 664–675, 2016. PMID: 26669825.

[66] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," *Phys. Rev. B*, vol. 87, p. 184115, May 2013.

[67] A. Bartók-Pártay, *The Gaussian Approximation Potential: An Interatomic Potential Derived from First Principles Quantum Mechanics*. Springer Theses, Springer Berlin Heidelberg, 2010.

[68] M. Rupp, R. Ramakrishnan, and O. A. von Lilienfeld, "Machine learning for quantum mechanical properties of atoms in molecules," *The Journal of Physical Chemistry Letters*, vol. 6, no. 16, pp. 3309–3313, 2015.

[69] M. Griffiths, S. P. Niblett, and D. J. Wales, "Optimal alignment of structures for finite and periodic systems," *Journal of Chemical Theory and Computation*, vol. 13, no. 10, pp. 4914–4931, 2017. PMID: 28841314.

[70] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 32, pp. 922–923, Sep 1976.

[71] E. A. Coutsias, C. Seok, and K. A. Dill, "Using quaternions to calculate RMSD," *Journal of Computational Chemistry*, vol. 25, no. 15, pp. 1849–1857, 2004.

[72] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, pp. 325–340, Dec 1987.

[73] A. Sadeghi, S. A. Ghasemi, B. Schaefer, S. Mohr, M. A. Lill, and S. Goedecker, "Metrics for measuring distances in configuration spaces," vol. 139, pp. 184118–184118, Nov 2013.

[74] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, Feb 1992.

[75] H. Li and R. Hartley, "The 3D-3D registration problem revisited," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, Oct 2007.

[76] J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *2013 IEEE International Conference on Computer Vision*, pp. 1457–1464, Dec 2013.

[77] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: pose estimation and correspondence," *Pattern Recognition*, vol. 31, no. 8, pp. 1019 – 1031, 1998.

[78] A. Mikhalev and I. V. Oseledets, "Rectangular maximum-volume submatrices and their applications," *ArXiv e-prints*, Feb. 2015.

[79] F. Le Gall, "Powers of Tensors and Fast Matrix Multiplication," *arXiv e-prints*, Jan 2014.

[80] J. R. Bunch and J. E. Hopcroft, "Triangular factorization and inversion by fast matrix multiplication," *Mathematics of Computation*, vol. 28, no. 125, pp. 231–236, 1974.

[81] J. L. Bentley, D. F. Stanat, and E. H. Williams, "The complexity of finding fixed-radius near neighbors," *Information Processing Letters*, vol. 6, no. 6, pp. 209 – 212, 1977.

[82] A. Cortinovis and D. Kressner, "Low-rank approximation in the Frobenius norm by column and row subset selection," *arXiv e-prints*, Aug 2019.

[83] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang, "Matrix approximation and projective clustering via volume sampling," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, SODA '06, (USA), p. 1117–1126, Society for Industrial and Applied Mathematics, 2006.

[84] L. V. Foster, "Rank and null space calculations using matrix decomposition without column interchanges," *Linear Algebra and its Applications*, vol. 74, pp. 47 – 71, 1986.

[85] K. Mehlhorn and S. Näher, "Dynamic fractional cascading," *Algorithmica*, vol. 5, pp. 215–241, Jun 1990.

[86] B. Chazelle, "A functional approach to data structures and its use in multidimensional searching," *SIAM Journal on Computing*, vol. 17, no. 3, pp. 427–462, 1988.

[87] "Synopsys QuantumATK 2019.12." https://www.synopsys.com/silicon/quantumatk.html.

[88] S. Smidstrup, T. Markussen, P. Vancraeyveld, J. Wellendorff, J. Schneider, T. Gunst, B. Verstichel, D. Stradi, P. A. Khomyakov, U. G. Vej-Hansen, M.-E. Lee, S. T. Chill, F. Rasmussen, G. Penazzi, F. Corsetti, A. Ojanperä, K. Jensen, M. L. N. Palsgaard, U. Martinez, A. Blom, M. Brandbyge, and K. Stokbro, "QuantumATK: an integrated platform of electronic and atomic-scale modelling tools," *Journal of Physics: Condensed Matter*, vol. 32, p. 015901, oct 2019.

[89] J. Schneider, J. Hamaekers, S. T. Chill, S. Smidstrup, J. Bulin, R. Thesen, A. Blom, and K. Stokbro, "ATK-ForceField: a new generation molecular dynamics software package," *Modelling and Simulation in Materials Science and Engineering*, vol. 25, p. 085007, Oct. 2017.

[90] G. L. Kellogg and P. J. Feibelman, "Surface self-diffusion on Pt(001) by an atomic exchange mechanism," *Phys. Rev. Lett.*, vol. 64, pp. 3143–3146, Jun 1990.

[91] X. W. Zhou, R. A. Johnson, and H. N. G. Wadley, "Misfit-energy-increasing dislocations in vapor-deposited CoFe/NiFe multilayers," *Phys. Rev. B*, vol. 69, p. 144113, Apr 2004.

[92] R. Street, "Hydrogen diffusion and electronic metastability in amorphous silicon," *Physica B: Condensed Matter*, vol. 170, no. 1, pp. 69 – 81, 1991.

[93] V. L. Deringer, N. Bernstein, A. P. Bartók, M. J. Cliffe, R. N. Kerber, L. E. Marbella, C. P. Grey, S. R. Elliott, and G. Csányi, "Realistic atomistic structure of amorphous silicon from machine-learning-driven molecular dynamics," *The Journal of Physical Chemistry Letters*, vol. 9, no. 11, pp. 2879–2885, 2018. PMID: 29754489.

[94] R. Murty and H. Atwater, "Empirical interatomic potential for Si-H interactions," *Physical review. B, Condensed matter*, vol. 51, pp. 4889–4893, 03 1995.

[95] "Go-ICP algorithm, git commit 937f114590f7df8b003d05f594b55527e230fef0." https://github.com/yangjiaolong/Go-ICP.

[96] "Go-Permdist algorithm, git commit 4db8e51a4719a405cc2db5475d2cef5453da68a3." https://github.com/matthewghgriffiths/fastoverlap.

[97] "libAtoms.org data repository." http://www.libatoms.org/Home/DataRepository.

[98] A. P. Bartók, J. Kermode, N. Bernstein, and G. Csányi, "Machine learning a general-purpose interatomic potential for silicon," *Phys. Rev. X*, vol. 8, p. 041048, Dec 2018.

[99] E. Rakhmanov, E. B. Saff, and Y. Zhou, "Minimal discrete energy on the sphere," 1994.

[100] "EON: Long timescale dynamics." https://theory.cm.utexas.edu/eon/index.html.

[101] G. Henkelman and H. Jónsson, "Long time scale kinetic Monte Carlo simulations without lattice approximation and predefined event table," *The Journal of Chemical Physics*, vol. 115, no. 21, pp. 9657–9666, 2001.

List of Figures

1.1 Formation of larger defects from smaller ones. 2
1.2 Polycrystalline silicon - grain boundaries are dark grey. 3

2.1 Double well potential. 14

3.1 Representation of the evolution of an impurity atom as a MJP. 16
3.2 Harmonic approximations in one dimension. 18
3.3 Flowchart of the KMC algorithm. 20
3.4 Flowchart of the on-the-fly KMC algorithm. 21
3.5 Modified potential energy in a hyperdynamics simulation. 22

4.1 A number of ad-atoms (purple) on a crystal surface. 23
4.2 Active regions (marked by white circles) in a particle system. 25
4.3 An atomic neighbourhood of an atom in a larger system. 28
4.4 Flowchart of a k-ART simulation. 29
4.5 Adjacency graphs for different cutoff radii. 32
4.6 Two structurally different atomic neighbourhoods with the same adjacency graph. 32
4.7 Structure of the local transition database. 58
4.8 Setup of a local transition search for a single neighbourhood. 59
4.9 Setup of local transition searches in multiple neighbourhoods. 60

5.1 The exchange mechanism on a platinum surface. 64
5.2 Initial state of the grain boundary simulation. 65
5.3 Platinum ad-atom: difference in local and global solutions. 66
5.4 Cutoff study for the amorphous silicon problem. 67
5.5 Cutoff study for the grain boundary problem. 68
5.6 Distribution of the wall times in the silicon dataset. 71
5.7 Extreme neighbourhoods in the silicon dataset. 72
5.8 Distribution of the wall times in the amorphous dataset. 72
5.9 Distribution of the wall times in the diamond dataset ($r = 6$ Å). 73
5.10 Relation between d^{full} and the runtimes for the diamond data set. 73
5.11 Distribution of the wall times in the sphere dataset ($n = 128$). 74
5.12 Worst-case wall times for the diamond example. 75

5.13 Worst-case wall times for the sphere example. 76
5.14 Average wall times for the silicon example. 77
5.15 Average wall times for the 128-particle sphere example. 77
5.16 Average wall times for the diamond example ($r = 6$ Å). 78
5.17 Ad-atom movement: number of new atomic environments in each KMC step. 79
5.18 Amorphous silicon problem: number of new atomic environments in each KMC step. 80
5.19 Two neighbourhoods that the nauty-based comparison determines to be almost equal. 80
5.20 Amorphous silicon problem: distribution of maximal particle movement during a transition. 81
5.21 Grain boundary problem: number of new atomic environments in each KMC step. 82
5.22 Example: MJP with missing transitions. 83
5.23 Ad-atom problem: median wall times. 84
5.24 Amorphous silicon problem: median wall times. 85
5.25 Amorphous silicon problem: relative errors in the total escape rates. 86
5.26 Grain boundary problem: relative errors in the total escape rates. 87
5.27 Grain boundary problem: median wall times. 88

List of Tables

4.1 Components of the complexity analysis. 54

A.1 AKMC/LKMC parameters for the ad-atom simulation in subsection 5.4.1. . . 92
A.2 AKMC/LKMC parameters for the hydrogen diffusion in amorphous silicon in subsection 5.4.2. 92
A.3 AKMC/LKMC parameters for the grain boundary simulation in subsection 5.4.3. 92

List of Algorithms

1 Algorithm that checks whether $f(N, \tilde{N}) = 0$. 41
2 Algorithm that checks whether $f(N, \tilde{N}) \leq \epsilon^2$. 42
3 Algorithm $comp^{full}(N, \tilde{N}, \epsilon)$ that checks whether $d(N, \tilde{N}) \leq \epsilon$. 43