Benjamin Welfler

Using PyRETIS to study electron transfer reactions between Ru(II)/Ru (III) in an aqueous solution

Master's thesis in applied theoretical chemistry
Supervisor: Titus Sebestiaan van Erp
Co-supervisor: Daniel Tianhou Zhang
June 2022

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

Benjamin Welfler

Using PyRETIS to study electron transfer reactions between Ru(II)/Ru (III) in an aqueous solution

Master's thesis in applied theoretical chemistry
Supervisor: Titus Sebestiaan van Erp
Co-supervisor: Daniel Tianhou Zhang
June 2022

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

NTNU
Norwegian University of
Science and Technology

Acknowledgements

This master thesis was conducted at the Department of Chemistry with Titus Sebastiaan van Erp as supervisor and Daniel Tianhou Zhang as co-supervisor. I would like to extend my deepest gratitude to both Titus and Daniel for all the assistance I have been given throughout this project, both personal and professional. This work would be impossible to complete without your aid. To my family, and my friends Ellen, Emilie, Herman, Kyrre, Maren, Mari, Ragnhild, Ruben, Thoa and Vårin, I thank you for all support throughout my study period. You have all kept me going, even when everything appeared bleak, which I will be forever grateful for.

Abstract

In this thesis, *ab initio* molecular dynamics (AIMD) simulations of the electron self-exchange reaction between Ru(II) and Ru(III) in an aqueous solution were used in combination with the path sampling method RETIS from the PyRETIS library. The development of simulation parameters for self-exchange reaction, study of the reaction mechanisms, and calculation of the rate constant was performed for a system with a pair of Ru(II) and Ru(III) ions, 63 water molecules and 1 OH^- molecule. The simulation cell was a cubic box of length 12.4138 Å with periodic boundary condition. From these simulations, we found that the position of the OH^- molecule, spatial alignment of the solvent, the OH bond distance for the OH^- molecule, and the amount of water complexes in the system were important factors for the reaction.

Through these simulations we added three criteria for the simulation of this system. These criteria were based on the formation of water complexes. The first criterion demanded there were no water complexes present in the system in the reactant or product state. The second criterion rejected any reaction pathway in which there were six or more water complexes at any given point. The third criterion rejected a path if it contained more than one water complex for 150 MD time steps. The development of a fourth criterion was started as a result of these simulations. Some trajectories contained phase points in which the self-consistent field (SCF) did not converge, resulting in a peak in the energy. The fourth criterion is suppose to recalculate the electronic configuration if the SCF does not converge, or reject the trajectory if the SCF is impossible to converge.

The rate constant for the self-exchange reaction was calculated to be
$k_{\mathrm{AB}} = 3.121 \cdot 10^{-4}$ [1/fs] with a relative error of 129.7%, as a proof of concept for the possibilities of rare event simulation.

Sammendrag

I denne avhandlingen ble *ab initio* molecular dynamics-simuleringer (AIMD) av elektron selvutvekslingsreaksjonen mellom Ru(II) og Ru(III) i en vandig løsning utført sammen med *rare-event path sampling* metoden RETIS fra PyRETIS biblioteket. Utviklingen av simuleringsparametere for selvutvekslingsreaksjonen, undersøkelsen av reaksjonsmekanismene og beregningen av hastighetskonstanten ble gjennomført for et system med et par Ru(II) og Ru(III) ioner, 63 vannmolekyler og 1 OH^- molekyl. Simulasjonsboksen var en kubisk boks med lengde 12,4138 Å og periodiske grensebetingelser. Fra disse simuleringene fant vi ut at posisjonen for OH^- molekylet, romlig orientering av løsemiddelet, OH bondlengden i OH^- molekylet og mengden vannkomplekser i system var viktige faktorer for reaksjonen.

Gjennom disse simuleringene har vi lagt til tre kriterier for simuleringen av systemet. Disse kriteriene var basert på dannelsen av vannkomplekser. Det første kriteriet krevde at det ikke var noen vannkomplekser tilstede i systemet i reaktant- eller produkttilstanden. Det andre kriteriet avviste enhver reaksjonsvei der det dannet seg seks eller flere vannkomplekser. Det tredje kriteriet avviste en reaksjonsvei om den inneholdt mer enn ett vannkompleks i over 150 simuleringstidspunkter. Utviklingen av et fjerde avvisningskriteriet ble igangsatt som et resultat av disse simuleringene. Noen reaksjonsveier inneholdt fasepunkter der det selvkonsistente feltet (SCF) ikke konvergerte, som resulterte i en topp i energien. Det fjerde kriteriet skal beregne den elektroniske konfigurasjonen på nytt hvis SCFen ikke konvergerte for fasepunktet, og eventuelt avvise reaksjonsveien hvis det er umulig å konvergere SCFen.

Hastighetskonstanten for elektronoverføringen ble beregnet til å være
$k_{AB} = 3.121 \cdot 10^{-4}$ [1/fs] med en relativ feil på 129,7%, som en demonstrasjon av mulighetene til rare-event simulering.

Contents

Introduction

Electron transfer reactions are one of the most elementary of all chemical reactions. They are the fundamental processes of redox reactions and play a vital part in areas such as novel energy sources,[1] DNA UV-damage repair,[2] metal corrosion, organic synthesis, energy storage devices,[3] etc. Electron transfer reactions occur so quickly, they are not even detected by the stopped-flow technique.[4] Electron self-exchange reactions are a special case where the reactant state and product state are the same. This makes them even more difficult to study experimentally, as there are no changes in concentration in the system.

To understand the kinetics of electron transfer reactions, we must employ computational methods. Using *ab initio* molecular dynamics,[5] in which the Schrödinger equation is solved using density functional theory for every MD step, it is possible to simulate the mechanisms of an electron transfer. Self-exchange reactions are activated processes, meaning they must cross a free energy barrier to go from reactant state to product state. This poses a challenge when using brute force AIMD, as the time step must be short enough to guarantee stable trajectories, without wasting computational resources simulating the equilibrium states.

Because AIMD simulations are expensive, we want to simulate the system during the electron transfer. With the introduction of transition path sampling (TPS) it is possible to generate an ensemble of reactive trajectories.[6] This is done by the use of Monte Carlo moves, where we make random changes (momenta, positions, etc.) in a previously accepted path to generate a starting point for a new trajectory. The new path will then explore the electron transfer, rather than the equilibrium states.

The TPS method was improved on by transition interface sampling (TIS), which introduced variable path lengths and interfaces between the stable states.[7] With these interfaces, we could now sample trajectories that do not end in the product state, but rather crosses a given interface for the corresponding ensemble. TIS was further improved on to create replica exchange transition interface sampling (RETIS).[8] This method introduces a $[0^-]$ ensemble to explore the reactant state for quicker rate calculations, as well making it possible to swap valid paths between ensembles.

The PyRETIS library[9,10] is an open source Python library for rare event simulation. This uses methods based on RETIS and TIS, and introduces some advanced Monte Carlo moves such as the *wire fencing*[11] move. By using PyRETIS with CP2K, the self-exchange reaction between Ru(II) and Ru(III) in aqueous solution was studied in this paper.

1 Theory

In this section, theoretical background for the mechanisms studied in this project, computer simulation methods and path sampling algorithms will be presented.

1.1 Reaction mechanisms

1.1.1 Order parameter

To study a chemical reaction we use an *order parameter* (OP), which can distinguish between the initial and final states of the reaction. OP can be a single variable, such as bond distance or a bond angle in a molecule, or it could be a collective variable (CV) such as the potential energy in a system.[12] It is a cumbersome task to to choose a functional OP that can be used to describe and enforce reactions. The choice of OP determines the quality of information obtained from the system, as a “bad” OP can count non-reactive pathways as reactive.[6] A priori knowledge of the system is therefore useful when choosing the order parameter.

1.1.2 Proton transfer reactions

Proton transfer reactions are reactions where a proton (H^+) is donated from one molecule to another, such as acid-base reactions. It is also possible that a water

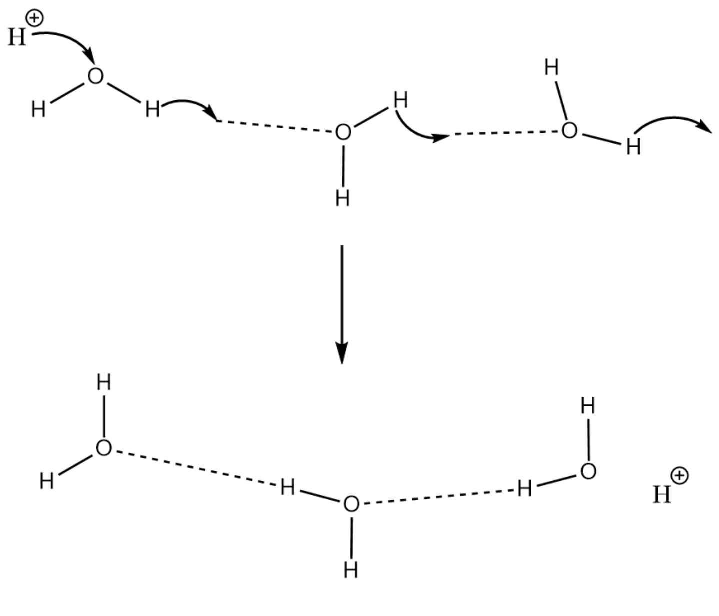

Figure 1.1: Reaction mechanism for a chain of proton transfer reactions in water known as the Grotthuss mechanism[13]

molecule forms a bond with a new proton and breaks one of the previous OH-bonds. This transfer can happen in a chain reaction known as the Grotthuss mechanism, in which several water molecules go through this proton transfer,[13] as seen in figure 1.1. These chain reactions of proton transfer enables the H^+ charge of the system to move more rapidly, compared to translational diffusion of a lone proton.[14]

1.1.3 Self-exchange electron transfer reaction

Electron transfer reactions are reactions involving an electron *donor* (D) and an electron *acceptor* (A). The general form is given as

$$D^{l+} + A^{(m+1)+} \to D^{(l+1)+} + A^{m+} \tag{1.1}$$

where l and m indicate the oxidation states through the reaction. Electron transfer reactions are mainly reorganization of the electron density. According to Marcus theory of electron transfer, the solvent and bonds of a system changes drastically before and after an electron transfer. During the electron transfer the atoms and molecules are practically immobilized compared to how fast the electron is transferred from the donor to the acceptor. For this reaction to occur, the solvent must therefore align in such a way that the reactant and product states are degenerate at the moment of the electron transfer for the energy to be conserved.[15,16]

Electron transfer reactions with two identical elements are known as a self-exchange reaction. This is given as

$$M^{n+} + M'^{(n+1)+} \to M^{(n+1)+} + M'^{n+} \tag{1.2}$$

where M and M' are the same elements with different oxidation states. Computational chemistry allows us to study properties in the transition state, as well as reaction intermediaries where it is impossible to study them experimentally. Self-exchange reactions are especially difficult to study experimentally as the reactant state and product state, leading to no concentration change over time. That is why we need computational simulation to understand the mechanisms during the electron transfer.[8,17] In computer simulations we require a way to obtain the energy of the system.

1.2 Electronic structure theory

The energy of a system is obtained through electronic structure theory. From quantum mechanics, the energy for a particle is given as the eigenvalues when operating the Hamiltonian operator on a particle's wave function as shown in equation (1.3), which is the time-independent Schrödinger equation(SE).[18]

$$\hat{H}|\Psi\rangle = E|\Psi\rangle \tag{1.3}$$

The $\hat{H}$ is the Hamiltonian operator, Ψ is the wave function and E is the energy of the particle. The Hamiltonian for for an N-electron, M-nuclei system is given as

$$\hat{H} = \hat{T}_{\text{e}} + \hat{T}_{\text{n}} + \hat{U}_{\text{en}} + \hat{U}_{\text{ee}} + \hat{U}_{\text{nn}} \tag{1.4}$$

where $\hat{T}_e$ and $\hat{T}_n$ are the kinetic energy of the electrons and nuclei and $\hat{U}_{en}$, $\hat{U}_{ee}$ and $\hat{U}_{nn}$ are the Coulomb interactions (attraction and repulsion) between electrons and nuclei in the system. To simplify our energy expression it is possible to use the Born-Oppenheimer (BO) approximation.[19] This assumes that electrons move faster than the nuclei, allowing us to treat the nuclei as stationary with electrons moving around them. This gives a multiplicatively separable wave function

$$\Psi_{\text{tot}} = \Psi_{\text{nuclei}}\Psi_{\text{electrons}} \tag{1.5}$$

and an additive separable energy

$$E_{\text{tot}} = E_{\text{nuclei}} + E_{\text{electrons}} \tag{1.6}$$

The Hamiltonian for the electronic wave function is then given by

$$\hat{H}_{\text{electrons}} = \hat{T}_{\text{e}} + \hat{U}_{\text{en}} + \hat{U}_{\text{ee}} \tag{1.7}$$

Because the nuclei are so slow compared to the electrons, the energy contribution from $\hat{T}_{\text{n}}$ is negligible compared to $\hat{T}_{\text{e}}$. Another consequence of the slow movement of the nuclei is that the Coulomb interaction between the nuclei, U_{nn}, is essentially constant on the timescale at which electrons move. It is then possible to solve the SE with only the electronic wave function.[20]

Analytical solutions to equation (1.3) only exist for a few single-electron systems.[21] In all other cases, we need to rely on numerical methods, in which the most accurate ones will also take the largest computational cost. Numerical methods, such as full-configuration interaction (full CI) or couple cluster (CC) can find the electronic structure for molecules at the size of N_2[22] (10 electrons) or $C_{150}H_{302}$ (452 atoms).[23] As the CC solution is only for one time frame, it is necessary to use other methods to obtain the energy of larger systems.

1.2.1 Force fields

By using *force fields*, we can obtain the potential energy as a function of the nuclear coordinates. This includes the Coulombic interaction and implicitly the electronic energy. In force fields, the electrons are considered as a part of the atoms, therefore all bonds must be defined explicitly, since the bonds are not determined from a solution of the SE. Force fields also neglect the quantum behavior of the nuclei, and treat them as classical particles.[24]

There are a wide variety of force fields such as the Lennard-Jones potential, where the potential energy is given as

$$U_{\mathrm{LJ}}(r) = 4\epsilon\left(\frac{\sigma}{r}^{12} - \frac{\sigma}{r}^{6}\right) \tag{1.8}$$

where ϵ is the interaction energy between two particles, σ is the distance where the intermolecular forces are zero and r is the distance between the centers of each particle. Other notable force fields are the Morse potential,[25] the Buckingham potential,[26] CHARMM,[27] OPLS[28] and Amber.[29] Because force fields forego solving the Schrödinger equation, they calculate the energy faster than methods that solve the SE.

The downside of force fields is their lack of flexibility. They require a large training set to accurately fit the empirical parameters, which is resource demanding. If there are no existing force fields for the system being studied, the choice is to either train a model through electronic structure calculation, or use the closest possible force field. Another challenge with force fields is the modelling of unexpected events. Because the force fields are based on empirical data, they accurately model the data which they have been trained by. Events such as bond breaking and forming make it difficult to parameterize the forces, because bonds have to be predefined.[24]

ReaxFF and "Empirical Valence Bond" (EVB) are force fields which aim to model breaking and forming of bonds. EVB was inspired by Marcus theory's calculation of the probability of electron transfer.[30] ReaxFF employs a bond order formalism, which allows for simulation of chemical bonds without quantum mechanics calculations.[31] A study in 2015 performed on silica dimerization reactions compared two ReaxFF parameter sets to density functional theory. Both ReaxFF force fields generated an unphysical formation of sodium hydroxide, which shows that reactive force fields might not be a viable method for obtaining the energy of a system with bond breaking and forming.[32]

To study the mechanisms of electron transfer reaction, it would be impossible to use a force field such as ReaxFF. A computational tool within the ReaxFF framework, called eReaxFF has been developed to simulating electrons.[33] The electrons are treated in a *pseudoclassical* manner.[34] This means that they move as particles with coordinates affected by interaction with other particles, however, the interactions between particles take quantum behavior into account. This allows for simulations that are several magnitudes faster than quantum chemistry methods.

With these types of force fields it could be possible to use force fields to simulate self-exchange reactions some time in the future, however, a large data set is still required for parameterization of the forces.[35] This is why quantum mechanical methods are required to study electron transfer reactions.

1.2.2 Density functional theory

Density functional theory (DFT) is a quantum mechanical modelling method which is able to simulate systems up to 1000 atoms.[36] DFT was created to forego solving the wave function by finding the ground-state electron density. The basis is formed from two theorems posed by Hohenberg-Kohn (HK). The first theorem states that the ground state electron density is a unique functional of the external potential,[37] which gives

$$E_0(\rho_0) = \int \rho_0(r) U_{\text{ext}} + F(\rho_0) \tag{1.9}$$

where E_0 is the ground state energy, ρ_0 is the ground state electron density, U_{ext} is the external potential affecting the electrons and $F(\rho)$ is the HK functional, given by

$$F(\rho_0) = T(\rho_0) + U_{\text{ee}}(\rho_0) \tag{1.10}$$

which expresses equation (1.7) in terms of the ground state electron density.

The second theorem says that the electron density which minimizes the total energy is the true ground state energy.[37] HK therefore has shown that there is a correlation between the electron density and ground state energy. We also know that the functionals $T(\rho_0)$ and $U_{\text{ee}}(\rho_0)$ exist, however we lack a mathematical expression for these functionals.[38]

Kohn-Sham (KS) presented a way to approximate these functionals. They introduced an ansatz to the exact ground state electron density, as an imaginary system of non-interacting particles with their own ground state energy functional.[39] Thus allowing the energy functional to be expressed as

$$E_0(\rho_0) = T_{\text{ref}}(\rho_0) + \int \rho_0(r) U_{\text{ext}} dr + E_H(\rho_0) + E_{\text{xc}}(\rho_0) \tag{1.11}$$

where $T_{ref}(\rho)$ is the kinetic energy of the reference system which is given from

$$T_{\text{ref}}(\rho) = -\frac{1}{2} \sum_{i}^{N} \langle \psi_i | \nabla^2 | \psi_i \rangle \tag{1.12}$$

and $E_H(\rho)$ is the Coulomb energy expressed as

$$E_{\mathrm{H}}(\rho) = \frac{1}{2}\int\int \frac{\rho(r)\rho(r')}{r-r'} dr dr' \tag{1.13}$$

and $E_{\mathrm{xc}}(\rho)$ is the exchange-correlation energy. The error approximations made by the imaginary system is presented as the exchange-correlation energy. By implementing an accurate E_{xc}, DFT should in principle give the exact electron density and total energy for any interacting and correlated system. How to calculate this exchange-correlation energy has been an ongoing problem since the 1980s, which started out as a model from Kohn-Sham called local density approximation (LDA), expressed as

$$E_{\mathrm{xc}}^{\mathrm{LDA}} = \int e_{\mathrm{XC}}^{\mathrm{UEG}}(\rho) \tag{1.14}$$

where $e_{\mathrm{XC}}^{\mathrm{UEG}}(\rho)$ is the exchange-correlation energy for each volume unit in a uniform electron gas.[40–42] Continuous development has been made to make the exchange-correlation energy as accurate as possible, as the accuracy of the DFT method is determined by it. For the past two decades the most used E_{xc} functional has been B3LYP (Becke, 3-parameter, Lee-Yang-Parr), which is a hybrid functional. B3LYP is derived by using a linear combination of the BLYP[43,44] and local spin density approximation (LSDA) functional.[42] The exchange-correlation energy for B3LYP is expressed as

$$E_{\mathrm{xc}}^{\mathrm{B3LYP}} = 0.8E_{\mathrm{x}}^{\mathrm{LSDA}} + 0.2E_{\mathrm{x}}^{\mathrm{HF}} + 0.72\Delta E_{\mathrm{x}}^{\mathrm{B}} + 0.19E_{\mathrm{c}}^{\mathrm{LSDA}} + 0.81E_{\mathrm{c}}^{\mathrm{LYP}} \tag{1.15}$$

where the constant factors are optimized values, $E_{\mathrm{x}}^{\mathrm{LSDA}}$ is the exchange energy functional from the local spin density approximation (LSDA), $E_{\mathrm{x}}^{\mathrm{HF}}$ is the Hartree-Fock exchange energy functional,[20] $E_{\mathrm{x}}^{\mathrm{B}}$ is the Becke 88 exchange energy functional,[43] $E_{\mathrm{c}}^{\mathrm{LSDA}}$ is the correlation energy functional from LSDA and $E_{\mathrm{c}}^{\mathrm{LYP}}$ is the correlation energy functional from Lee-Yang-Parr.[44,45]

For the simulations performed in this thesis we used the BLYP functional, which uses Becke's proposed gradient correction to the functional. This corrects the LDSA result to

$$E_{\mathrm{x}}[\rho(\mathbf{r})] = E_{\mathrm{x}}^{\mathrm{LSDA}}[\rho(\mathbf{r})] - b\sum_{\sigma=\alpha,\beta}\int \rho_\sigma^{\frac{4}{3}} \frac{x_\sigma^2}{1+6bx_\sigma sinh^{-1}x_\sigma} d\mathbf{r};\ x_\sigma = \frac{|\nabla\rho_\sigma|}{\rho_\sigma^{\frac{4}{3}}} \tag{1.16}$$

where $E_{\mathrm{x}}[\rho(\mathbf{r})]$ is the correlation energy and b is a constant with the value of 0.0042 a.u. BLYP uses the LYP functional as the correlation functional. This is given as

$$E_{\mathrm{c}}[\rho(\mathbf{r})] = -a\int \frac{\gamma(\mathbf{r})\rho(\mathbf{r})}{1+d\rho(\mathbf{r}}\left\{1 + \frac{3}{10}2^{\frac{5}{3}}b(3\pi^2)^{\frac{2}{3}}\left[\frac{\rho_\alpha}{\rho}^{\frac{8}{3}} + \frac{\rho_\beta}{\rho}^{\frac{8}{3}}\right]e^{-c\rho^{-\frac{1}{3}}}\right\}d\mathbf{r} \tag{1.17}$$

where $\gamma(\mathbf{r})$ is given as

$$\gamma(\mathbf{r}) = 2\bigg(1 - \frac{\rho_\alpha^2(\mathbf{r}) + \rho_\beta^2(\mathbf{r})}{\rho(\mathbf{r})}\bigg) \tag{1.18}$$

and $a = 0.049$, $b = 0.132$, $c = 0.2533$ and $d = 0.349$.[24,46]

Because the second HK theorem requires that the true electron density is the one that minimizes the energy of the system, we can solve this with a variational approach. This is done by finding the lowest energy eigenstate, which corresponds to the ground state. By formulating a Kohn-Sham operator

$$\hat{h}_{\text{KS}} = \hat{h}_{\text{KS}}\psi_i^{(n)} \tag{1.19}$$

and solving the Kohn-Sham orbital equation

$$\Big(-\frac{1}{2}\nabla^2 + U_{\text{KS}}\Big)\psi_i = \epsilon_i\psi_i \tag{1.20}$$

where U_{KS} is given from

$$U_{\text{KS}} = U_{\text{ext}} + U_{\text{el}} + \frac{\delta E_{\text{xc}}}{\delta\rho} \tag{1.21}$$

where $\frac{\delta E_{\text{xc}}}{\delta\rho}$ is the functional derivative of the exchange-correlation energy with respect to the electron density. By solving

$$\hat{h}_{\text{KS}}^{(n)}\psi_i^{(n+1)} = \epsilon_i^{(n+1)}\psi_i^{(n+1)} \tag{1.22}$$

we can compare the wave functions $\psi_i^{(n)}$ to $\psi_i^{(n+1)}$. If the difference between the two iterations are lower than a predefined tolerance, then we have reached *self-consistency*.[38]

1.3 Molecular dynamics

Molecular dynamics (MD) is a molecular simulation method where we determine the movement of particles in our system by integrating Newton's second law of motion

$$\frac{\partial^2 x_i}{\partial t^2} = \frac{F_{x_i}}{m_i} \tag{1.23}$$

where m_i is the mass of particle i, x_i is the position of particle i along a coordinate and F_{x_i} is the force force affecting the particle in the given direction.[45]

Despite the increase in computational power, MD simulations typically consists of between 10^2 to 10^5 atoms with a time scale of 10^{-9} to 10^{-6} seconds. A large scale

MD simulation was ran for a biological process containing 1.6 billion atoms had a performance of 8.30 ns/day.[47] It is therefore impossible to run MD simulations on macroscopic systems, which consists of somewhere around 10^{23} atoms. However, we can use methods such as *periodic boundary conditions* to approximate an infinitely large system to study bulk properties in our systems.[48]

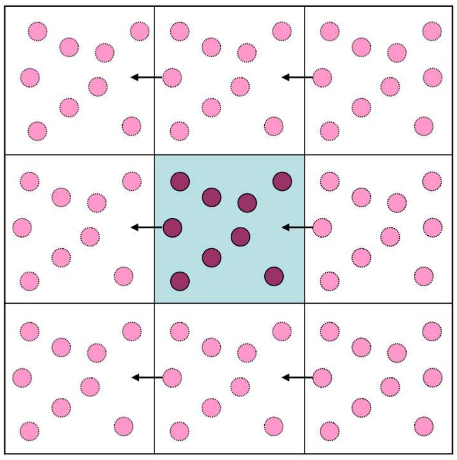

Figure 1.2: A two dimensional illustration of periodic boundary conditions. The original system is shown in the middle square, are infinitely duplicated in all directions as shown by the surrounding squares.[49]

This method infinitely replicates the system in all directions of the simulation box as seen in figure 1.2. If a particle leaves the simulation cell, it reenters in the opposite side of the cell, keeping the amount of particles constant. This approximation allows us to compare small model systems with laboratory experiments.

There are many parameters to decide for an MD simulation. Size and shape of simulation box, time step integrator, length of each time step and the prescribed potential for the initial configuration. Choosing too long of a time step can cause interactions such as particles landing on top of each other. However a too short time step is resource demanding and will not explore the entire phase space. Appropriate time steps for flexible molecules and bonds with translational, rotational, torsional and vibrational motion are between 0.1 and 1 fs (10^{-16} to 10^{-15} seconds).[24]

1.3.1 Ab initio molecular dynamics

With classical MD the prescribed potential make certain processes difficult to simulate. As mentioned previously, bond breaking and forming is difficult, even when using ReaxFF. To accurately compute the forces of the particles at a given time step we can use quantum mechanics to calculate their electronic structure. This is referred to as *ab initio* molecular dynamics (AIMD).[5,50] Ab initio being Latin for "from the beginning" implies that the Schrödinger equation is solved for every MD step.

Another benefit with AIMD is the Car-Parrinello (CP) extended Lagrangian approach.[51] While the KS orbitals were modeled for a system of non-interacting particles, CPMD introduces a time dependence for their orbitals. This means we can regard the orbitals as evolving in space and time similarly to those in relativistic quantum field theories.[52] These orbitals with minimum spatial spread are known as "Wannier orbitals".[53]

For electron transfer reactions, this allows us to observe the "movement" of these Wannier orbitals. These movements can be used to understand the mechanisms for electron transfer reactions. Due to computational limitations, we currently cannot run AIMD simulations with numerically accurate methods such as full CI or CC. Because electronic structure calculations are expensive, DFT is currently the most dominant method.

1.4 Metropolis Monte Carlo

Metropolis Monte Carlo (MC) is a stochastic method in which the algorithm performs random changes to the system's properties, such as changing the position of the particles in the system randomly. The potential energy of the newly generated configuration is then calculated. This new configuration is then either accepted or rejected. The acceptance probability is given by

$$P_{\text{acc}}(o \to n) = MIN\left[1, \frac{e^{-\beta U(\mathbf{r}_{\text{new}})}}{e^{-\beta U(\mathbf{r}_{\text{old}})}}\right] \tag{1.24}$$

where $U(\mathbf{r})$ is the potential energy for the system denoted by the old and new configuration, β is the reciprocal temperature[54] given as $\frac{1}{k_B T}$ where k_B is the Boltzmann constant and T is the temperature. If the energy is higher in the new configuration, a random float (q) between 0 and 1 is generated and compared to the Boltzmann distributed energy.[55] The move is accepted if:

$$q \leq e^{-\beta \Delta U(\mathbf{r})} \tag{1.25}$$

Because of its stochastic nature, the algorithm can sample higher energy states with a representative probability distribution using the law of large numbers. The scheme for the Metropolis MC algorithm can be seen in figure 1.3.

It must be *Markovian*, meaning that each trial only depend on the preceding trial. Additionally, each trial must belong to a finite set of possible outcomes.[56]

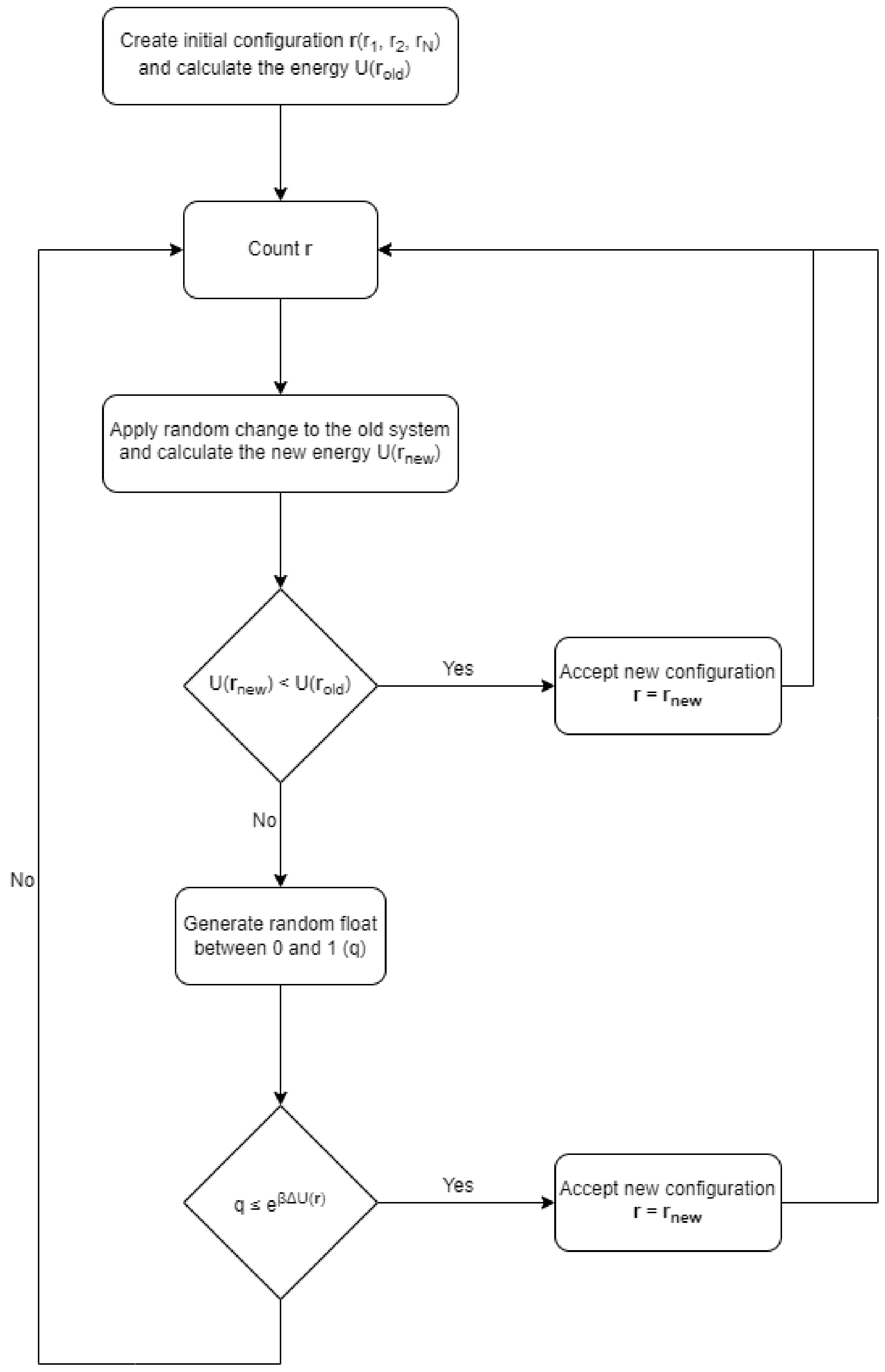
Create initial configuration $\mathbf{r}(r_1, r_2, r_N)$ and calculate the energy $U(r_{old})$
Count r
Apply random change to the old system and calculate the new energy $U(r_{new})$
$U(\mathbf{r}_{new}) < U(r_{old})$
Yes
Accept new configuration $\mathbf{r} = \mathbf{r}_{new}$
No
Generate random float between 0 and 1 (q)
$q \leq e^{\beta \Delta U(\mathbf{r})}$
Yes
Accept new configuration $\mathbf{r} = \mathbf{r}_{new}$
No

Figure 1.3: Schematic representation of the Metropolis Monte Carlo algorithm

All locations in the conformation space needs to be accessible (*ergodic*).[57] The Markov chain of the system also needs to obey *detailed balance*. For a given move, this implies

$$P(o)\pi(o \to n) = P(n)\pi(n \to o) \tag{1.26}$$

Where o and n represents the old and new states and π is the probability to access a state as a function of two states.[58]

1.5 Path sampling

Processes such as chemical reactions have more than one way to go from the reactant state (state A) to the product state (state B), such as different intermediaries or solvent structure. It is therefore important to sample an ensemble of these reactive *paths* to get an accurate understanding of the process. A *path* or *trajectory* is defined as an ordered chain of time slices as shown in equation (1.27).

$$\mathbf{X}(L) = \{\mathbf{x}_0(\mathbf{r}_0, \mathbf{p}_0), \mathbf{x}_1(\mathbf{r}_1, \mathbf{p}_1), \mathbf{x}_i(\mathbf{r}_i, \mathbf{p}_i)..., \mathbf{x}_L(\mathbf{r}_L, \mathbf{p}_L)\} \tag{1.27}$$

where $\mathbf{X}$ is the trajectory, L is the path length, $\mathbf{x}_i(\mathbf{r}_i, \mathbf{p_i})$ is the phase point, $\mathbf{r}_i(r_1, r_2, ..., r_N)$ and $\mathbf{p}_i(p_1, p_2, ..., p_N)$ are the coordinates and the momenta for a system with N particles, for the time slice i.[6] Path sampling is necessary to study so called *rare events*. A rare event is an event which occur infrequently compared to the other processes in the system.[8] A rare event can occur several times within a second and still be considered rare, because the rarity is compared to the other events in the system.

MD simulations that are set up to study chemical reactions spend most of the time in equilibrium as reactants or products, hence the occurrence of a reaction is considered rare. In addition to occurring with a low frequency, rare events also tend to be quick. This makes them difficult to study using only brute force MD, as the time step has to be short enough to guarantee stable trajectories, without wasting resources on simulating the equilibrium states. This is why the usage of path sampling is required for processes such as electron transfer reaction. Path sampling uses a combination of MC algorithm to make changes to a known reactive path, and then employing MD to generate a new trajectory. With this, it is possible to study the mechanisms of the process as efficiently as possible.[59]

1.5.1 Transition path sampling

Transition path sampling (TPS) is a path sampling technique which makes it possible to study rare events using MD simulation. This is done through the MC *shooting*, *time reversal* and *shifting* moves, in which a reactive trajectory is subject to a change, and then integrated forwards and backwards in time. Eventually this will explore the phase space of the chemical reaction with the ensemble of reactive trajectories.[60]

The algorithm behind the *shooting move* is that it selects a random phase point of an old reactive trajectory ($\mathbf{x}^{(o)}$) and "shoots" from it. This involves a random change in a property, such as position or momentum at the selected point as shown in figure 1.4.

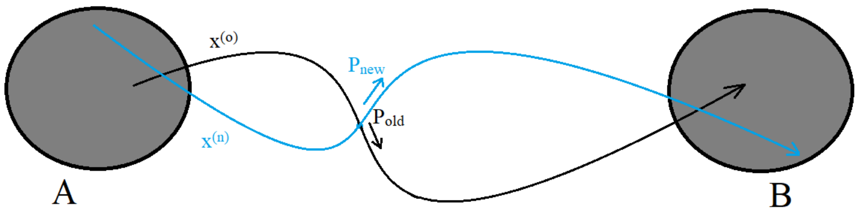

Figure 1.4: Graphical representation of the Monte Carlo shooting move. A random point in the phase space of a reactive trajectory ($\mathbf{x}^{(o)}$) between state A and state B is selected. The momenta of the particles in the original trajectory ($\mathbf{P}_{\text{old}}$) are then changed ($\Delta\mathbf{P}$) to obtain a new vector of momenta ($\mathbf{P}_{\text{new}}$). The system is then evolved forwards and backwards in time to generate a new trajectory $\mathbf{x}^{(n)}$

The old momenta ($\mathbf{P}_{\text{old}}$) is changed by a random amount ($\Delta\mathbf{P}$) to obtain a new vector of momenta ($\mathbf{P}_{\text{new}}$). The system is then evolved forwards and backwards in time using MD until the path starts in state A and ends in B.[61]

The *time reversal* move reverses an accepted path by changing the direction of the path, as illustrated in figure 1.5.[9]

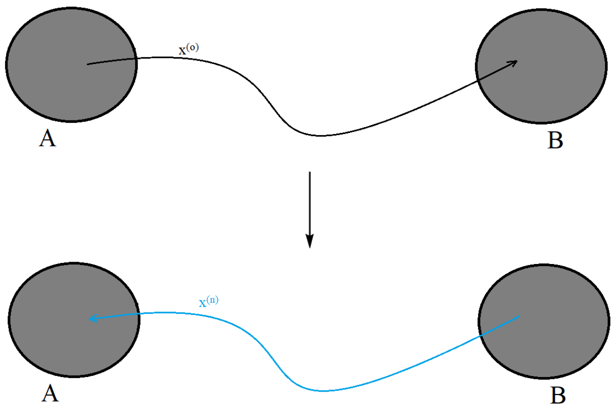

Figure 1.5: Graphical illustration of the time reversal move. The black path $\mathbf{x}^{(o)}$ is the old path starting in state A and ending in state B. The blue path $\mathbf{x}^{(n)}$ is the new path after the time reversal

The *shifting move* can be performed as *forwards shifting* or *backwards shifting*. In the forwards shifting, a random time slice δt from the start of the trajectory is selected. This time slice, as well as all slices before, are then removed from the trajectory, and the trajectory is then evolved δt forwards in time. The forwards shifting is shown in figure 1.6.

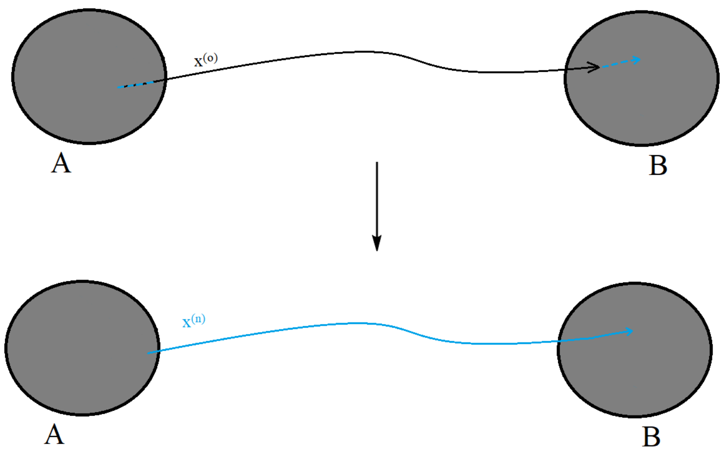

Figure 1.6: Graphical representation of the forwards shifting move. A random time slice, as well as all previous time slices are removed from the start of the trajectory. The trajectory is then evolved forwards in time by the same amount as the time slice which was removed.

The backwards shifting move is similar to the forwards shifting except a time slice at the end of the trajectory is selected and the system is evolved backwards in time.[60] This move has been made redundant, and removed with the introduction of transition interface sampling.

1.5.2 Transition interface sampling

Transition interface sampling (TIS) is an improvement of the TPS method designed to calculate reaction rates. The main differences between TIS and TPS is the implementation of ensembles, the removal of the shifting move and variable path lengths. The ensembles are a collection of paths which crosses a corresponding interface (λ_0, λ_1,..., λ_n). This allows for sampling of reactive and "almost reactive" trajectories, which makes it possible to study the difference between reactive and unreactive paths. The interfaces are placed such that $\lambda_{i-1} < \lambda_i$, as well as λ_0 and λ_n represent the states A and B.

The placement of the interfaces is illustrated in figure 1.7. The interfaces for states A and B are renamed λ_A and λ_B in this example. Figure 1.7 show how the trajectories move from state A and crossing these interfaces.[7]

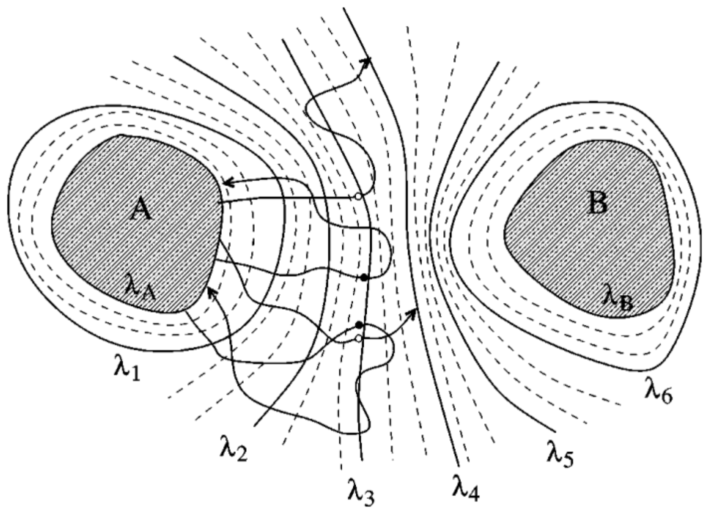

Figure 1.7: Illustration of the interfaces from λ_1 to λ_6, as well as λ_A and λ_B defining the stable states A and B. The arrows indicate trajectories' movement in the phase space[7]

Another benefit to TIS is the variable path length. In TPS the trajectories have a fixed length, meaning that even if the path goes from state A to B, the time evolution is still continued for the predetermined amount. TIS saves a lot of computational resources by only simulating the reaction, instead of the stable states. It is due to the variable path length that the shifting move is no longer necessary, as TPS uses it to increase the statistics.

TIS also introduces a way to calculate the rate constant by using the flux through the interfaces. For a rare event, the flux is impossible to calculate due to it being very small. However, the interfaces enables us to write the flux as crossing probabilities between the interfaces, giving us

$$k_{\mathrm{AB}} = f_{\mathrm{A}} P_{\mathrm{A}}(\lambda_{\mathrm{B}}|\lambda_{\mathrm{A}}) = f_{\mathrm{A}} \prod_{i=0}^{n-1} P_{\mathrm{A}}(\lambda_{i+1}|\lambda_i) \tag{1.28}$$

where f_{A} is the initial flux, which measures how often trajectories start off a λ_{A}. This is obtain through an MD simulation of the system. λ_{A} and λ_{B} refer to state A and B respectively, λ_i is a given interface and $P(\lambda_{i+1}|\lambda_i)$ is the conditional crossing probability of crossing the subsequent interface after crossing λ_i.[62]

1.5.3 Replica exchange transition interface sampling

Replica exchange transition interface sampling (RETIS) is an improvement of TIS. RETIS adds in a *swapping* move, in addition to using the same moves as TIS.[63]

This move swaps paths that are valid for different ensembles between themselves as shown in figure 1.8.

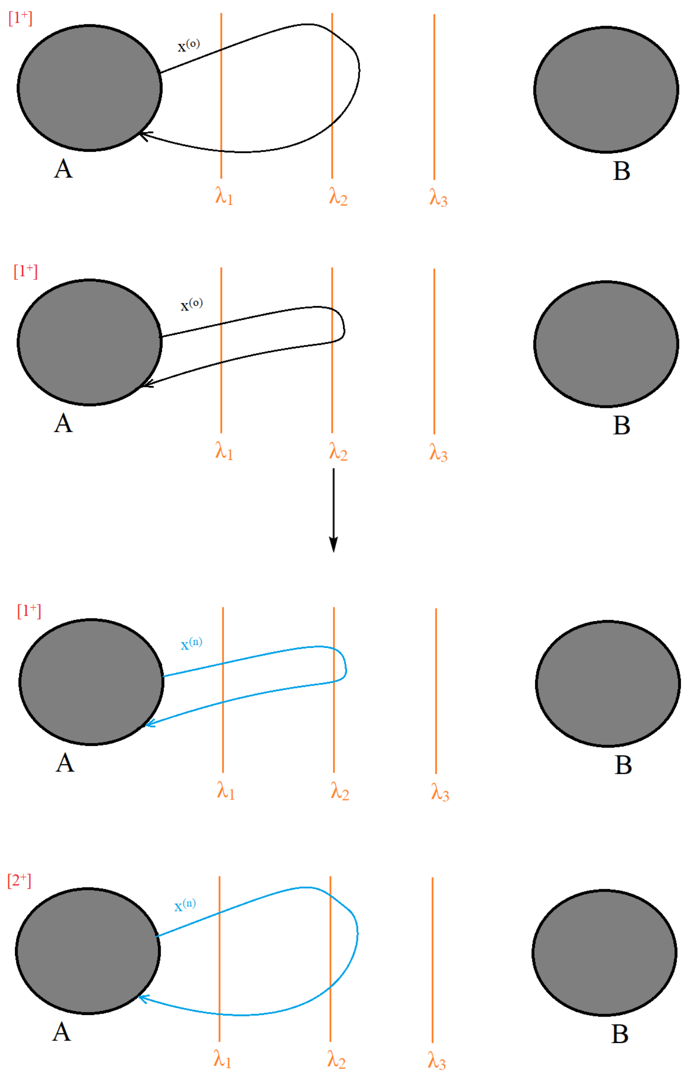

Figure 1.8: Graphical illustration of the swapping move between the $[1^+]$ and $[2^+]$ ensemble.

If a path in the $[3^+]$ ensemble also crosses the λ_4 interface, and a path in the $[4^+]$ ensemble crosses the λ_3 interface, these paths can be added to the other ensemble. It is a simple method which decrease correlation between paths within the ensembles.

For this paper, the Python library PyRETIS[9, 10] was utilized. This library uses methods based on TIS and RETIS. In addition to the previous moves, it uses a move called the *wire fencing* move.[11] When sampling the [i$^+$] ensemble, this move selects a phase point that is above λ_i on an old trajectory (black) as shown in figure 1.9.

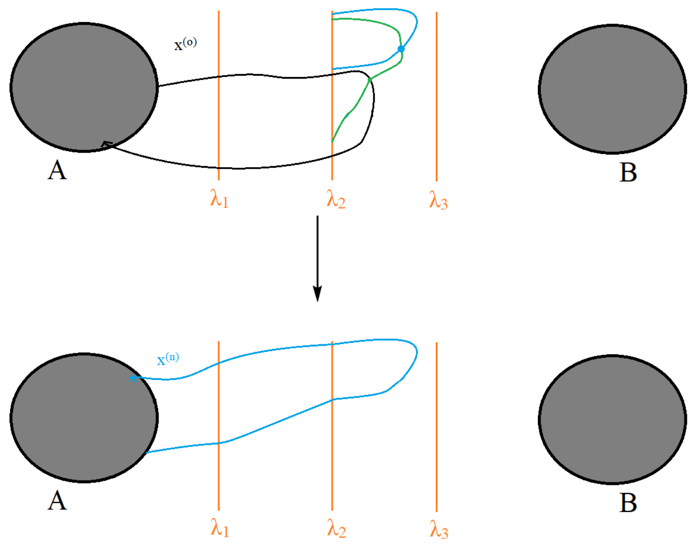

Figure 1.9: Graphical illustration of the wire fencing move between the stable states A and B for the ensemble [i$^+$] using two subpaths. A phase point above λ_i on the initial trajectory (black) $\mathbf{x}^{(o)}$ is selected. A new path (green) is generated that starts and ends at λ_i. From this trajectory a new phase point is selected and another trajectory is generated (blue). At the bottom the time evolved new path $\mathbf{x}^{(n)}$ is shown.

Using this phase point, a new trajectory (green) is generated that starts and ends at λ_i. A second subpath (blue) is generated by selecting a new phase point on the green path and performing a shooting move, then time evolving it until it starts and ends at λ_i. After generating the subpaths, the last accepted subpath is then time-evolved in both directions until it reaches λ_A or λ_B. If the path results in a B to B path, the move is rejected, otherwise it is accepted with the probability from equation (1.24).

This illustration is for a scheme using two subpaths, but it is possible to add more. The wire fencing move gives more decorrelated paths than a TPS shooting move. This is because the trajectories will share at least one phase point in a shooting move, while the wire fencing move could have no common phase points between the old and new path.

RETIS also introduces the [0$^-$] ensemble. This ensemble consists of trajectories which explore the reactant state. By exploring the reactant state, the initial flux can be calculated in RETIS as

$$f_{\mathrm{A}} = \frac{1}{\langle t^{[0^-]} \rangle + \langle t^{0^+} \rangle} \tag{1.29}$$

where $\langle t^{[0^-]} \rangle$ and $\langle t^{[0^+]} \rangle$ are the average path lengths for the $[0^-]$ and $[0^+]$ ensemble. With this $[0^-]$ ensemble, we no longer need to run an MD simulation to obtain the initial flux.[64]

2 Method

The simulations for this thesis were performed on the Idun HPC cluster[65] and on resources provided by Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway.

2.1 Model system

The system was a cubic box of 12.4138×12.4138×12.4138 Å with periodic boundary conditions. In the system there were 63 H_2O molecules, 1 OH^- molecule, 1 Ru(III) ion and 1 Ru(II) ion as shown in figure 2.1. Ewald summation was used, which provides a uniform background charge distribution that makes the simulation box neutral.[66]

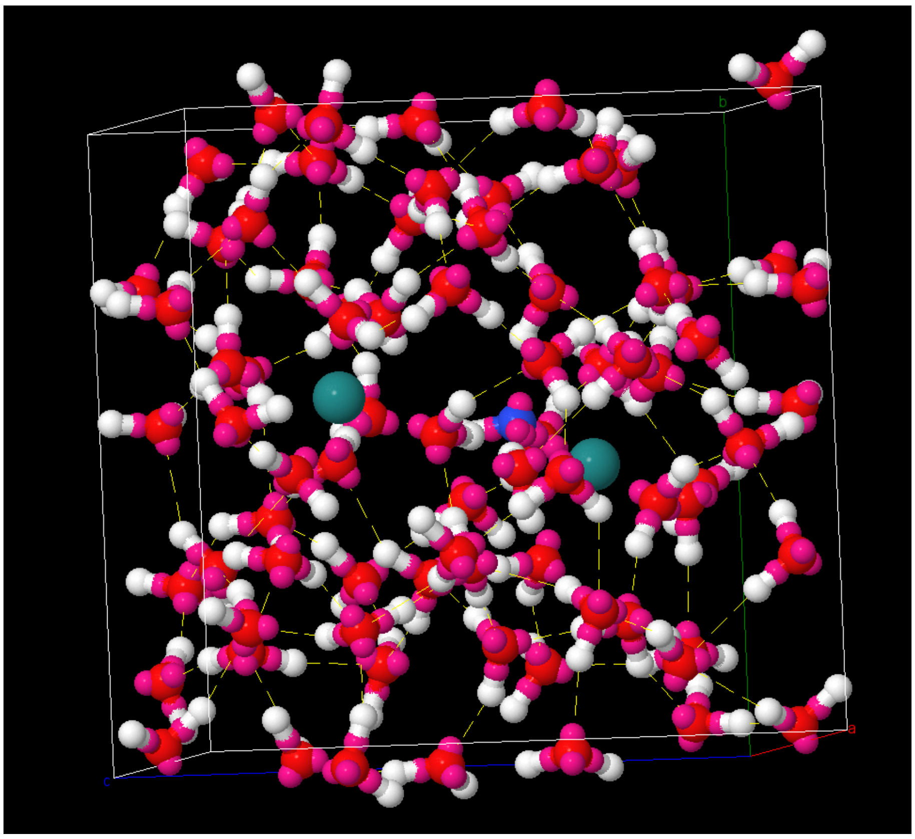

Figure 2.1: Visualization of the model system in Jmol. Ruthenium colored as green, oxygen as red, hydrogen as white, electron as pink and the oxygen in the OH^- molecule as blue.

2.2 Simulation setup

To generate the initial trajectory, an AIMD simulation was ran using the CP2K[67] engine with QUICKSTEP DFT and BLYP as exchange-correlation functional. The electronic structure is modeled using Gaussian and plane wave basis sets, the hydrogen and oxygen were modeled using DZVP-GTH basis and for the ruthenium DZV-GTH were used. After calculating the electronic structure, the KS-orbitals were projected on the Wannier orbitals to describe the "position" of the electron.

Based on this trajectory the PyRETIS library[9, 10] was utilized to generate new initial configurations, which were time integrated with the CP2K[67] engine.

2.2.1 Order parameter

The order parameter ξ was defined as

$$\xi_{\mathrm{ET}} = \frac{d_{\mathrm{Ru-X}} - d_{\mathrm{Ru'-X}}}{d_{\mathrm{Ru-Ru'}}} \tag{2.1}$$

where Ru and Ru′ denote Ru(II) and Ru(III) respectively. The denoted X is the maximally localized Wannier center "moving" from the Ru(II) to the Ru(III). The $d_{\mathrm{Ru-X}}$ is the distance between the Wannier center and Ru(II), $d_{\mathrm{Ru'-X}}$ is the distance between the Wannier center and Ru(III), $d_{\mathrm{Ru-Ru'}}$ is the distance between the Ru(III) and Ru(II). The choice of order parameter is based on the research by A. Tiwari and B. Ensing.[68]

The Wannier center that is determined as the order parameter is chosen by iterating through the atomic coordinates and assigning electrons until their "outer shell" is full. Meaning 6 electrons are assigned to each oxygen (7 for the oxygen in the OH^- molecule), 1 electron to each hydrogen and 4 electrons to each ruthenium. The order parameter is the Wannier center which is left after this iteration. From equation (2.1) this value is -1 at our defined state A and 1 at state B., and somewhere in between during the electron transfer.

3 Results and discussion

This was the first time RETIS was used for simulation of electron transfer reactions. Most of the work in this thesis is therefore focused on the developments made by me and my collaborators. This involves changes to the CP2K input files and code changes to the PyRETIS program, how it was implemented and why it was implemented. Because most of the work has been on the developments of how to simulate the electron transfer reaction, the final results are based on very few MC steps. These results will be presented along with suggestions for future work.

For each simulation, the accepted paths were visualized using Jmol, which is an open-source Java viewer for chemical structures in 3D.[69] Using Python to preprocess the data and adding the periodic boundary conditions enabled this visualization. These visualizations served as a qualitative study of the mechanisms during the transition.

A key factor to simulating electron transfer reactions is accurate placement of the interfaces. The crossing probability for a reaction $P_{\mathrm{A}}(\lambda_{\mathrm{B}}|\lambda_{\mathrm{A}})$ is fixed, however, by adding interfaces we obtain local crossing probabilities $P(\lambda_{(i+1)}|\lambda_i)$, which are much higher. We want to place these interfaces so that $P(\lambda_{(i+1)}|\lambda_i) \approx 0.2$, which is based on optimal usage of computer resources.[8]

If the interfaces are placed too close to each other we get a crossing probability $P(\lambda_{(i+1)}|\lambda_i) \approx 1$. This is undesirable as $\lambda_{(i+1)}$ provide the same information as λ_i, at the cost of simulation time. By placing the interfaces too far apart, the crossing probability becomes $P(\lambda_{(i+1)}|\lambda_i) << 1$. This means that the single crossing probabilities requires too much time to converge.

During these simulations we noticed the there were significant changes to the potential energy, and uneven movement of the Wannier center for a few phase points in some trajectories. These “jumps” were the results of the self-consistent field (SCF) not converging, due to extremely delocalized electronic structure occurring when the electron is moving between the two ruthenium ions. It is difficult to converge the SCF for these configurations, because there are competing, almost degenerate electronic states. If the SCF does not converge, the result is a peak in the total energy of the system, meaning the forces are discontinuous. This can lead to unintended bond breaking or forming, giving us an unphysical configuration.

The significant increases in the potential energy was therefore troublesome, as the PyRETIS program shoots from a previously accepted path. Consequentially, most subsequent trajectories had unphysical behavior. To avoid these types of paths we implemented three criteria to the solvent structure which will be presented later.

3.1 Changes in the simulation

Throughout this thesis, four PyRETIS simulations and one equilibrium simulation were ran. The first simulation was an equilibrium run to decide where to place the interfaces for the RETIS simulation. This system ran for 2202.5 fs using 4405 time steps as shown in figure 3.1.

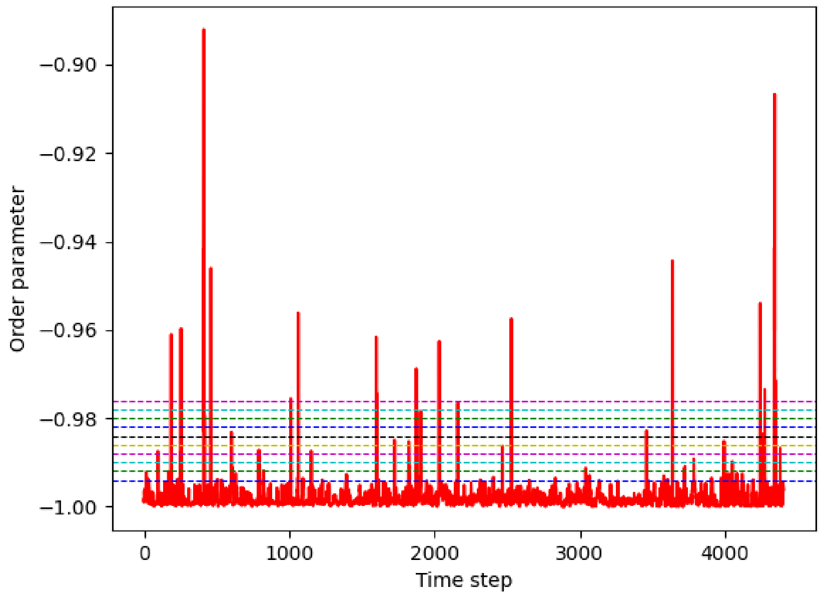

Figure 3.1: Order parameter (OP) plotted as a function of MD steps using the definition of OP from equation (2.1). The dashed lines show the placement of the interfaces [-0.994, -0.992, -0.990, -0.988, -0.986, -0.984, -0.982, -0.980, -0.978, -0.976]. The interface λ_B is placed at [0.994], however, it is omitted from the figure for visual purposes. These interfaces were used in the first PyRETIS simulation.

The interfaces for the first PyRETIS simulation were placed at [-0.994, -0.992, -0.990, -0.988, -0.986, -0.984, -0.982, -0.980, -0.978, -0.976, 0.994], as shown in figure 3.1. The 0.994 interface was omitted from figure 3.1, for the reader to be able to distinguish between the other interfaces. The simulation were then started using the wire fencing move with two subpaths and “high acceptance”. The high acceptance allows for paths that go from state B to state A by reversing the path so it starts in A and ends in B, which was introduced in PyRETIS’ *stone skipping* move.[63]

To improve on the convergence of the CP2K engine we separated the SCF parameter into an outer and inner loop. For the equilibrium run the max amount of SCF iterations were set to 150, which was changed to 30 inner loops and 5 outer loops. With the outer loops we make changes to the guess which can avoid situations where the simulation converge to an unphysical configuration.

The first PyRETIS simulation ran for 245 cycles giving the crossing probabilities from the simulation is shown in table 3.1.

Table 3.1: Crossing probabilities from the first PyRETIS simulation running for 245 cycles. λ_i and λ_{i+1} shows the placement of the interfaces necessary to cross for a given ensemble.

Ensemble	Crossing probability	λ_i	λ_{i+1}
$[0^+]$	0.796380	-0.994	-0.992
$[1^+]$	0.650778	-0.992	-0.990
$[2^+]$	0.917582	-0.990	-0.988
$[3^+]$	1.000000	-0.988	-0.986
$[4^+]$	1.000000	-0.986	-0.984
$[5^+]$	1.000000	-0.984	-0.982
$[6^+]$	1.000000	-0.982	-0.980
$[7^+]$	1.000000	-0.980	-0.978
$[8^+]$	1.000000	-0.978	-0.976
$[9^+]$	0.000227	-0.976	0.994

Using the crossing probabilities we notice that the interfaces 3-8 might be redundant, as their crossing probabilities are equal to 1. This means that every path crossing interface λ_i also crosses λ_{i+1}. The matched overall crossing probabilities based on the local crossing probabilities shown in figure 3.2 was also used to decide on the new interfaces.

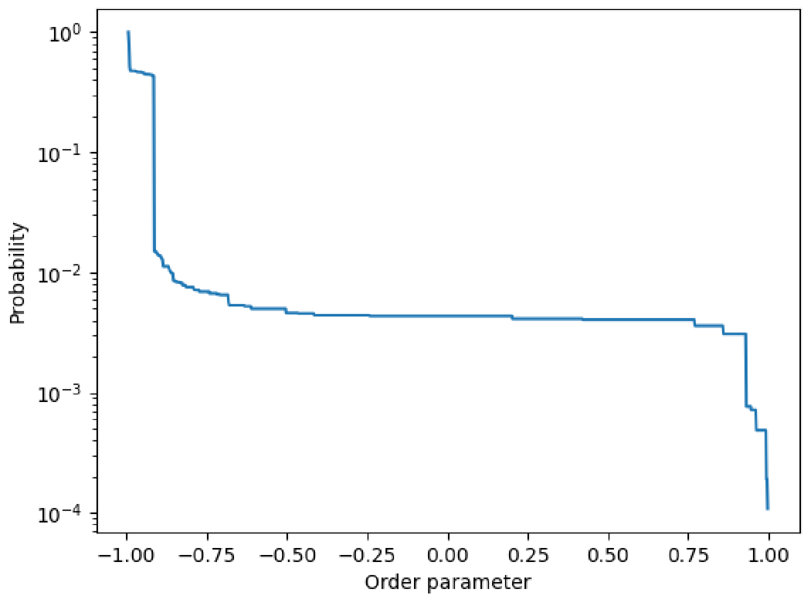

Figure 3.2: Matched overall crossing probabilities based on the local crossing probabilities from the first PyRETIS simulation.

Because 6/10 interfaces had 1 as the crossing probability we ran a new simulation to reduce the number of interfaces. The new interfaces were placed at [-0.994, -0.990, -0.986, -0.982, -0.978, 0.994]. This new simulation ran for 172 cycles and gave the crossing probabilities displayed in table 3.2.

Table 3.2: Crossing probabilities from the second PyRETIS simulation running for 172 cycles. λ_i and λ_{i+1} shows the placement of the interfaces necessary to cross for a given ensemble.

Ensemble	Crossing probability	λ_i	λ_{i+1}
$[0^+]$	0.408163	-0.994	-0.990
$[1^+]$	0.945818	-0.990	-0.986
$[2^+]$	0.601619	-0.986	-0.982
$[3^+]$	0.742506	-0.982	-0.978
$[4^+]$	0.002338	-0.978	0.994

The matched overall crossing probabilities based on the local crossing probabilities for this simulation is shown in figure 3.3.

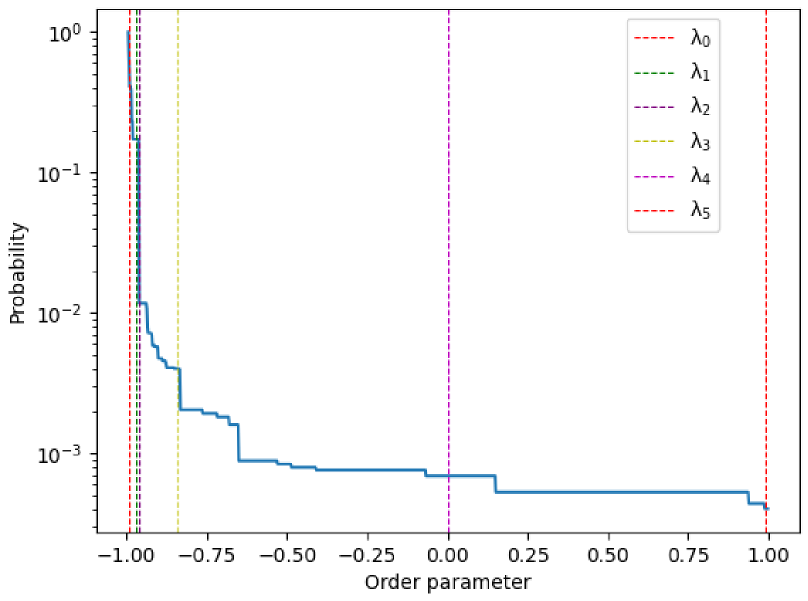

Figure 3.3: Matched overall crossing probabilities based on the local crossing probabilities from the second PyRETIS simulation. The placement of the interfaces for the third PyRETIS simulation are shown as dashed vertical lines.

The new interfaces chosen for the third PyRETIS simulation were [-0.990, -0.970, -0.960, -0.840, 0.000, 0.990] which are marked as colored vertical dashed lines in figure 3.3. These were selected for approximately 0.2 crossing probability between the interfaces from the second simulation.

The crossing probabilities for the third PyRETIS simulation is provided in table 3.3 from 197 cycles.

Table 3.3: Crossing probabilities from the third PyRETIS simulation running for 197 cycles. λ_i and λ_{i+1} shows the placement of the interfaces necessary to cross for a given ensemble.

Ensemble	Crossing probability	λ_i	λ_{i+1}
$[0^+]$	0.641176	-0.990	-0.970
$[1^+]$	0.708600	-0.970	-0.960
$[2^+]$	0.294180	-0.960	-0.840
$[3^+]$	0.020824	-0.840	0.000
$[4^+]$	0.986289	0.000	0.990

From table 3.3 we notice the crossing probability $P(\lambda_4|\lambda_3)$ which is 0.020824. Because of this, we want to investigate the area between these two interfaces. The matched overall crossing probabilities based on the local crossing probabilities from the third PyRETIS simulation shown in figure 3.4.

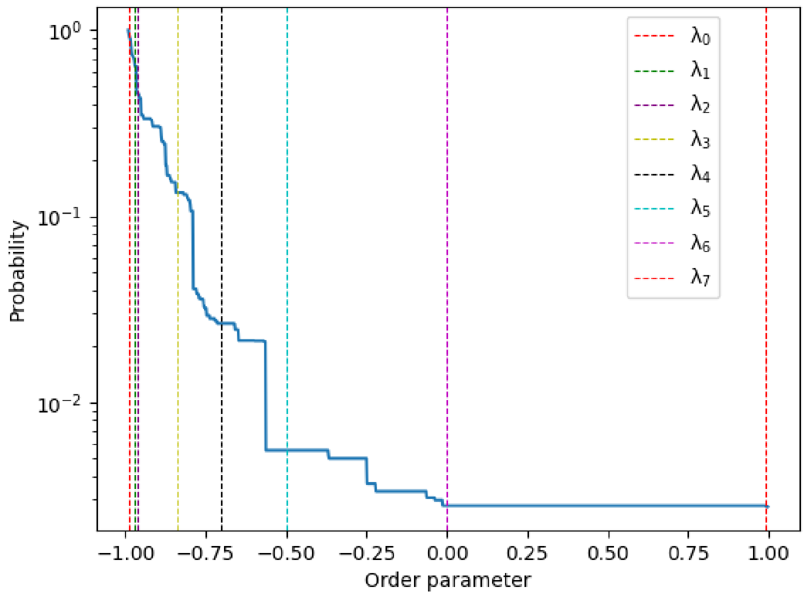

Figure 3.4: Matched overall crossing probabilities based on the local crossing probabilities from the third PyRETIS simulation. The placement of the interfaces for the fourth PyRETIS simulation are shown as dashed vertical lines.

For the fourth PyRETIS simulation the interfaces λ_4 and λ_5 was added at -0.700 and -0.500 respectively as shown in figure 3.4.

The crossing probabilities after 114 cycles from the fourth PyRETIS simulation is shown in table 3.4. The crossing probabilities for the $[1^+]$ and the $[4^+]$ ensemble sticks out with 1 and 0.995491 respectively. From previous iterations these could then either be moved or removed. However, as seen in table 3.3, the location of the interface for the $[1^+]$ ensemble is the same, but they differ with ≈ 0.3 in crossing probability. The interface placed at 0.000 has a 0.986289 crossing probability from the third simulation, while it has a crossing probability of 0.458777 from the fourth simulation. This would indicate that we might not have enough statistics to know if it is best to change them or not. By changing the interface prematurely, they might not yield the desired statistics.[61]

The changes in the crossing probabilities could also be due to three new stopping criteria added for the fourth simulation, which will be elaborated on later. This

Table 3.4: Crossing probabilities from the fourth PyRETIS simulation running for 114 cycles. λ_i and λ_{i+1} shows the placement of the interfaces necessary to cross for a given ensemble.

Ensemble	Crossing probability	λ_i	λ_{i+1}
$[0^+]$	0.100000	-0.990	-0.970
$[1^+]$	1.000000	-0.970	-0.960
$[2^+]$	0.499919	-0.960	-0.840
$[3^+]$	0.870069	-0.840	-0.700
$[4^+]$	0.995491	-0.700	-0.500
$[5^+]$	0.909243	-0.500	0.000
$[6^+]$	0.458777	0.000	0.990

could reject the trajectories containing the unphysical electonic configurations, which would be accepted in the third simulation. If the interfaces were to be moved based on the results in table 3.4, placing them around [-0.990, -0.970, -0.930, -0.860, 0.180, 0.570, 0.990] could be beneficial. This is illustrated in figure 3.5 with dashed lines at the suggested interfaces.

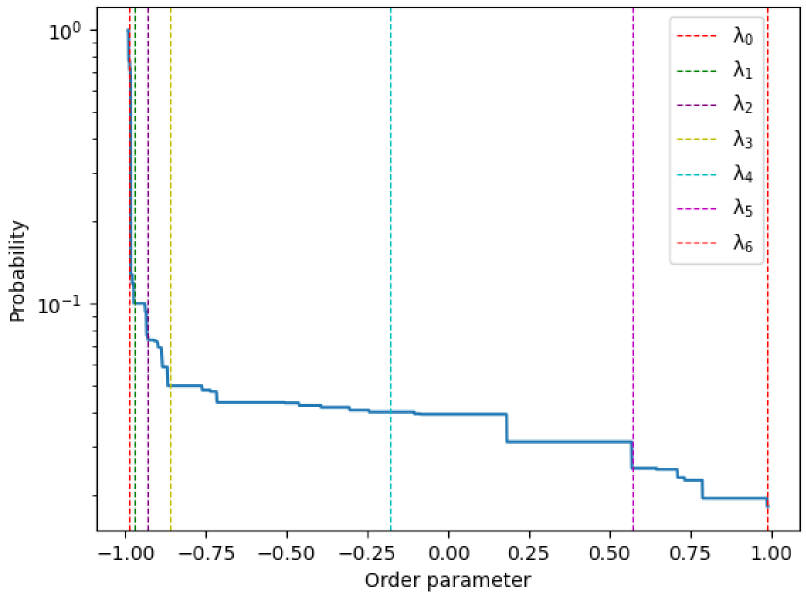

Figure 3.5: Matched overall crossing probabilities based on the local crossing probabilities from the fourth PyRETIS simulation. The suggested placement for a new PyRETIS simulation are shown as dashed vertical lines. These are based on limited statistics, therefore the next simulation should be continued with the current interfaces until more data is gathered.

These suggested interfaces were placed at locations where the probability drastically decreases. Because the sample size is low, and the probabilities from previous simulations indicate that we sample different areas of the phase space, it would be better to gather more data before these are moved.

As previously stated, three stopping criteria were added to the fourth iteration of the simulations. The first criterion was added to ensure that the system contained no water complexes when the trajectory is accepted. When the OP was in either state A ($\xi_{ET} \leq -0.990$) or state B ($\xi_{ET} \geq 0.990$), we iterate through every oxygen and create a bond using the bond distances from Jmol (0.91/2 Å for hydrogen and 1.81/2 Å for oxygen). For the stable system there should be 1 OH^- and 63 H_2O molecules. If the oxygen and hydrogen is bonded in any other way than the stable system, the OP was moved inside the barrier region, to ensure that the simulation continued. This new order parameter can be expressed as

$$\xi_{\mathrm{ET}} = \begin{cases} -0.989, & \text{if } N_{\text{complexes}} > 0 \text{ and } \xi_{\text{prev}} \leq -0.990 \\ 0.989, & \text{if } N_{\text{complexes}} > 0 \text{ and } \xi_{\text{prev}} \geq 0.990 \\ \frac{d_{\mathrm{Ru-X}} - d_{\mathrm{Ru'-X}}}{d_{\mathrm{Ru-Ru'}}}, & \text{otherwise} \end{cases} \tag{3.1}$$

The definition of the OP from equation (2.1) was still tracked to understand how the OP moved while the system was unstable.

A benefit with this criterion was avoiding paths with unstable conformations. From previous iterations we noticed paths which contained water complexes while in state A or B had one or more phase points where the SCF did not converge. As seen in figure 3.6, the phase point at time step 93 in trajectory 5 in ensemble [2^+] has a 13.01 Hartree increase in energy due to the SCF not being able to converge.

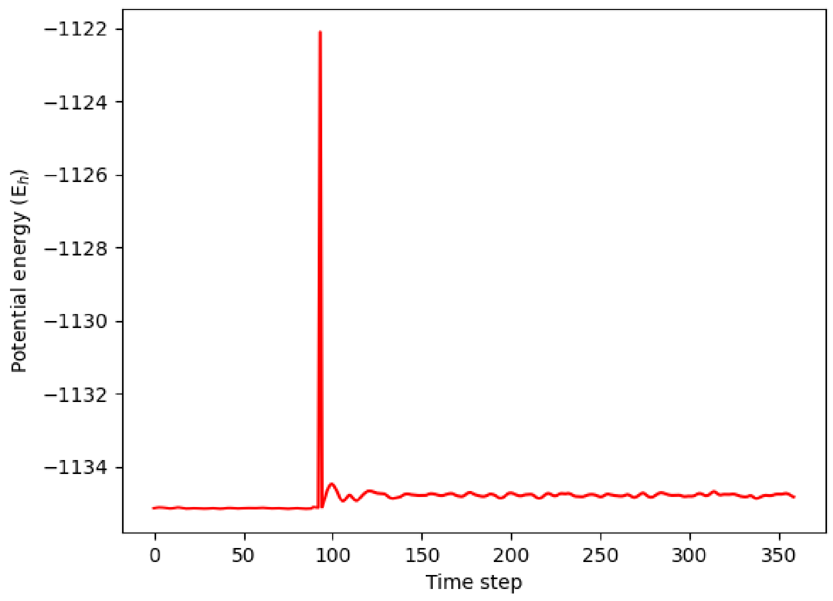

Figure 3.6: The potential energy (E_h) measured in Hartree as a function of time from trajectory 5 in the $[2^+]$ ensemble. The peak in energy at step 93 is an increase by 13.01 Hartree from the previous step. This energy increase is due to the SCF not converging, resulting in unphysical behavior.

The sudden energy increase gave unfavorable conformations for the water structure. After the energy jump the fluctuation of the energy is higher than before, which could indicate unphysical behavior in the system. By continuing the simulation while the water structure was unstable, these types of paths tended to get rejected. Because these paths required more SCF loops, they were more computationally expensive than the stable systems. The rejection rate increased, but we no longer got some trajectories which used more than one week to complete.

Another important aspect of the criterion was the studying of multiple recrossings of the simulation barrier region ($-0.989 \leq \xi_{\mathrm{ET}} \leq 0.989$). We wanted to investigate whether an unstable water structure could lead to the OP re-entering the simulation barrier region. As shown in figure 3.7, the OP moves from state A to state B, then exits state B after 70 time steps. In contrast, the OP from the equilibrium run in figure 3.1 does not leave state A within 4405 time steps. This indicates that the water structure has a significant effect on the self-exchange reaction, which is what we expect from the literature.[15, 16, 68, 70]

What we did not expect was how the order parameter moved while inside the

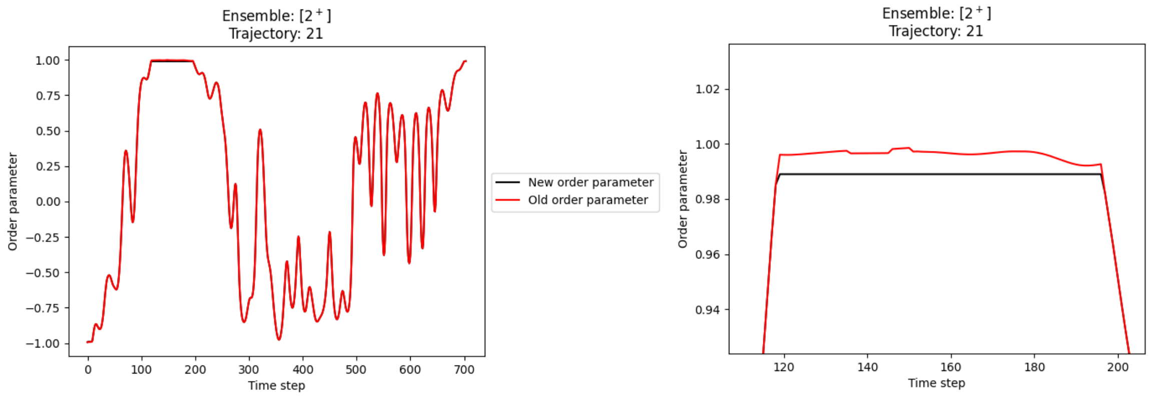

Figure 3.7: Trajectory 21 in the $[2^{+}]$ ensemble demonstrating recrossing of the simulation barrier region after implementing the new order parameter. The old parameter from equation (2.1) is shown as red and the new order parameter from equation (3.1) is shown in black. The graph to the right is zoomed in on the time steps 115 to 205 to display how the Wannier center moves within state B.

simulation barrier region. Because the electron move much faster than the nuclei, one would expect the order parameter to move straight from state A to B. However, we observed the OP moving forwards and backwards as seen in figure 3.9. This could be due to the Grotthuss mechanism which will be discussed later.

Because the simulations are resource-intensive, this criterion is inefficient, as phase points where the SCF did not converge gave several water complexes in the system. This led to a lot of CPU time being spent on continuing to simulate paths that would eventually get rejected. Therefore, this criterion should be kept if studying recrossings is desirable. The second stopping criterion in the fourth simulation was designed to minimize the resources wasted from the first criterion.

The second criterion is simply to reject any path where the system contains six or more water complexes at any point in the simulation. The amount of complexes were determined by data from previous simulations, where no path containing six or more complexes were ever accepted or possible. It would be possible to increase or reduce this number, but it was chosen to speed up the simulation along with the third criterion. The third criterion rejects any path where there are two or more complexes for the duration of 150 time steps. This amount is also chosen based on previous results, which could be subject to change.

As an improvement to the water complex criteria, a new method for recalculating and rejecting paths is being developed. By choosing a previous path which contained a sudden significant increase in energy and attempting to converge the SCF once more, the energy did not spike as shown in figure 3.8.

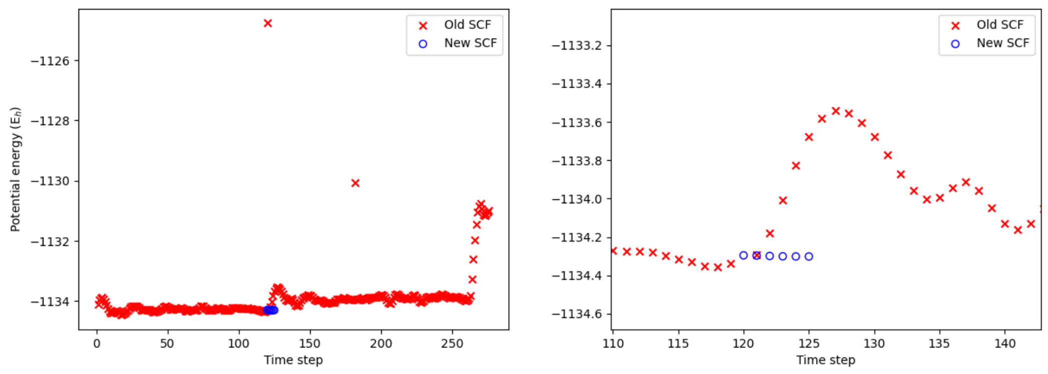

Figure 3.8: The potential energy (E_h) measured in Hartree of a trajectory shown as a function of the time steps. The red crosses are the potential energy from the original SCF calculations and the blue circles are the potential energy from the recalculated SCF. This displays how we were able to converge an electronic configuration, which previously did not converge. The plot to the right is a zoomed view of where the recalculations were performed.

The red crosses show the initial calculated potential energy where the SCF did not converge. The blue circles are our recalculations of the electronic structure until it converged. This illustrates that it is possible to converge the SCF for a configuration, which previously did not converge. Thereby proving that the energy jumps were caused by the SFC not converging. With the new criterion, paths with phase points where the SCF do not converge after recalculation are rejected when they occur. Although more data is required, this could replace the previous criterion for configurations containing more than six water complexes. This is due to the fact that all paths with more than six water complexes, also contained a phase point in which the SCF did not converge.

3.2 Reaction mechanics

The mechanics of the system was investigated throughout all four PyRETIS simulations using Jmol to visualize the system. Important factors for the self-exchange reaction was found to be the movement of the Wannier center throughout the simulation, the location of the OH^-, the spatial orientation of the water molecules and the formation of water complexes in the system.

3.2.1 Rate constant

The rate constant for the self-exchange reaction was calculated for the four PyRETIS simulations ran as shown in table 3.5. The drastic reduction of the relative error from the first and second to the third and fourth simulation is noticeable. Because

Table 3.5: Fluxes and rate constants for each iteration of the PyRETIS simulations with the corresponding relative error (%) for the rate constants.

Iteration	Flux [fs^{-1}]	Rate constant [fs^{-1}]	Relative error (%)
1	$2.67 \cdot 10^{-2}$	$2.888 \cdot 10^{-6}$	$5.076 \cdot 10^{2}$
2	$2.02 \cdot 10^{-2}$	$8.157 \cdot 10^{-6}$	$7.148 \cdot 10^{2}$
3	$1.50 \cdot 10^{-2}$	$4.128 \cdot 10^{-5}$	$1.479 \cdot 10^{2}$
4	$1.72 \cdot 10^{-2}$	$3.121 \cdot 10^{-4}$	$1.297 \cdot 10^{2}$

the rate constant calculation is dependent on the flux using equations (1.29) and (1.28), the rate constant from the final PyRETIS simulation has an innate error. A bug followed the change in the order parameter from equation (3.1), where the water structure is required to be stable while in state A or B. The simulation is continued by moving the OP to -0.989 if it is in state A, or to 0.989 if it is in state B if there are any water complexes in the system. For the [0^-] ensemble this poses an issue, because it explores the negative direction. With the current implementation of the criteria, if any water complexes form during a simulation in the [0^-] ensemble, the OP is moved to -0.989, which completes the trajectory and it is accepted, instead of continued until the system is stable.

3.2.2 Electron transfer

How the Wannier center moves between state A and B throughout the simulation has been a key parameter we have been studying.

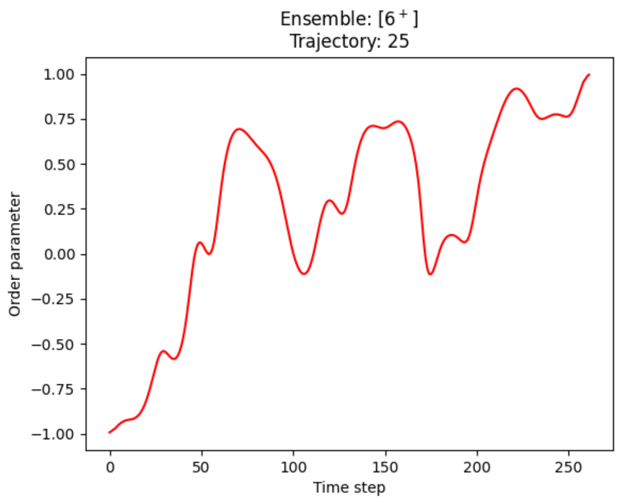

Figure 3.9: The order parameter as a function of time in trajectory 25 from the [6^+] ensemble. This illustrates the smooth movement of the Wannier center between state A and B. This path is visualized in figure 3.10.

Figure 3.9 is an example of a trajectory that moves from state A to state B. The movement of the OP is continuous through the simulation. Another notable observation is that the OP moves forwards and backwards. This implies that the system is volatile while the OP is inside the simulation barrier region ($-0.989 \leq \xi_{\mathrm{ET}} \leq 0.989$), which can be caused by the Grotthuss mechanism, allowing a chain of proton transfer reactions to occur during the electron transfer. The smooth movement of the OP in trajectory 25 from the $[6^{+}]$ ensemble can be seen in figure 3.10. The Wannier center (pink) move back and forth inside the simulation barrier region.

Despite this smooth movement of the OP, it is important to be critical of these trajectories. The trajectory shown in figure 3.9 corresponds to what we expect, but that should not be conclusive evidence. The path shown to the left in figure 3.11, has a more unexpected erratic movement.

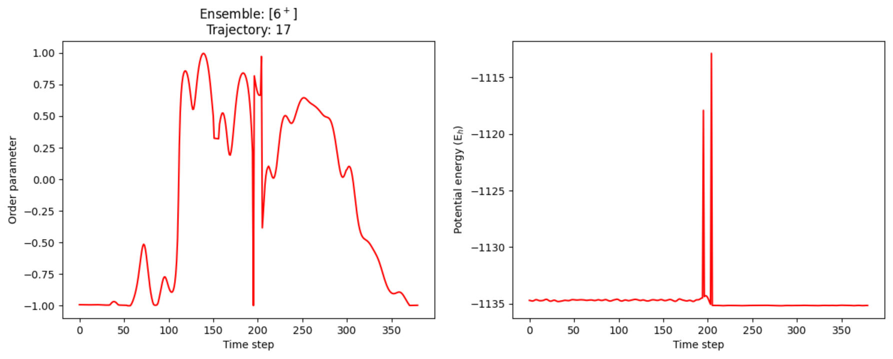

Figure 3.11: Plotted order parameter as a function of time (left) and potential energy ($\mathrm{E_h}$) measured in Hartree as a function of time (right) for trajectory 17 from the $[6^{+}]$ ensemble. We can see that the sudden changes in the order parameter corresponds to the sudden peaks in energy at the time steps 195 and 205. This shows how the SCF which did not converge can affect the OP. The jump at step 195 is shown in figure 3.12.

At time steps 195 and 205, we notice two large changes in the OP. It could be possible that these two states are degenerate, so how can we tell correct and incorrect paths apart? The right graph in figure 3.11 shows the potential energy measured in Hartree (E_h) as a function of time. The significant increases in energy at the time steps 195 and 205 corresponds to the rapid movement of the OP. This would indicate that this behavior is a result of a badly converged electronic configuration.

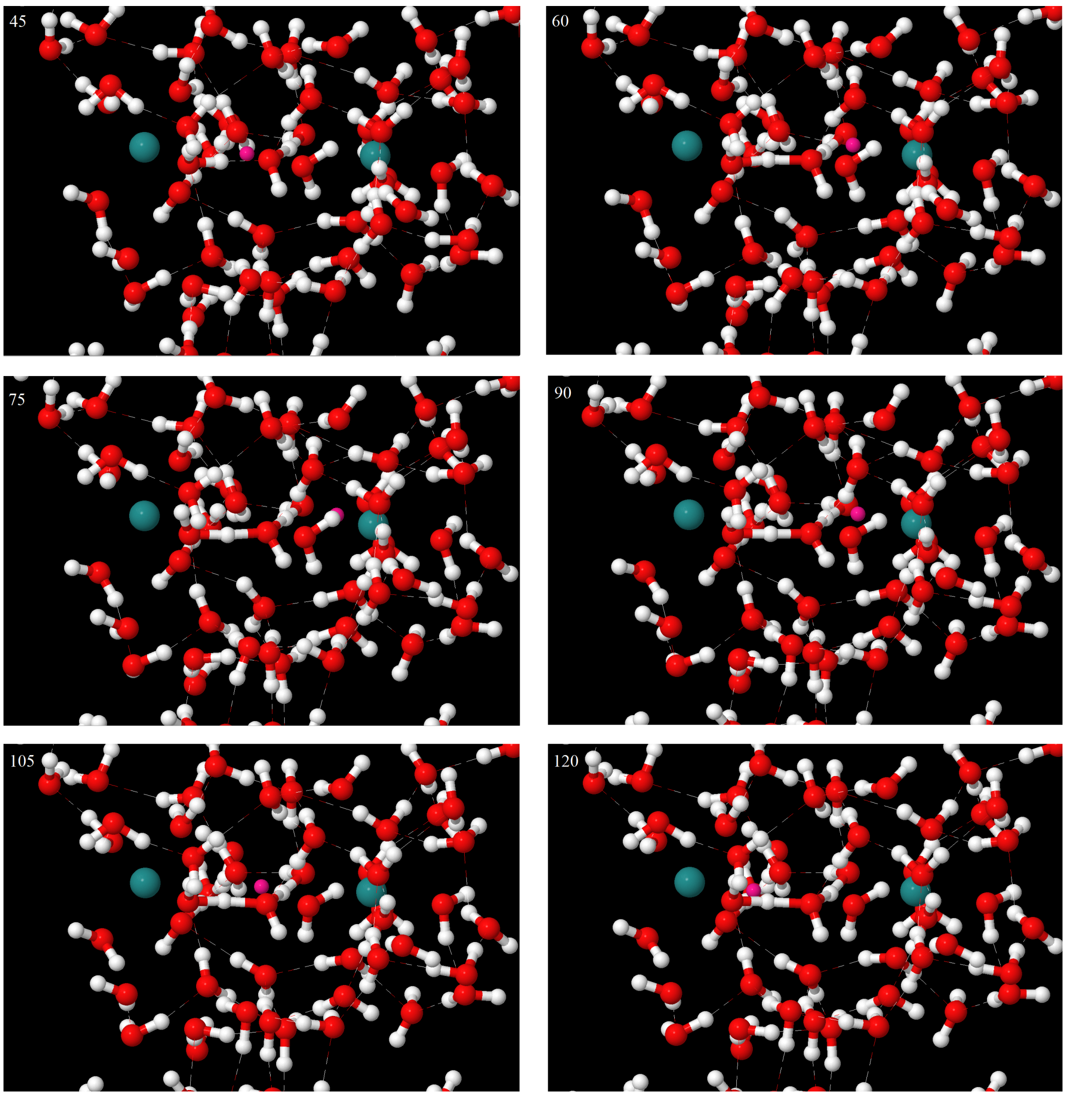

Figure 3.10: Visualized movement of the order parameter using Jmol at the time steps 45, 60, 75, 90, 105 and 120 in trajectory 25 in the [6^+] ensemble where the SCF converged for every point. Ruthenium is shown as green, oxygen as red, hydrogen as white and the Wannier center (OP) as pink. All electrons except the OP have been hidden for clearer illustration.

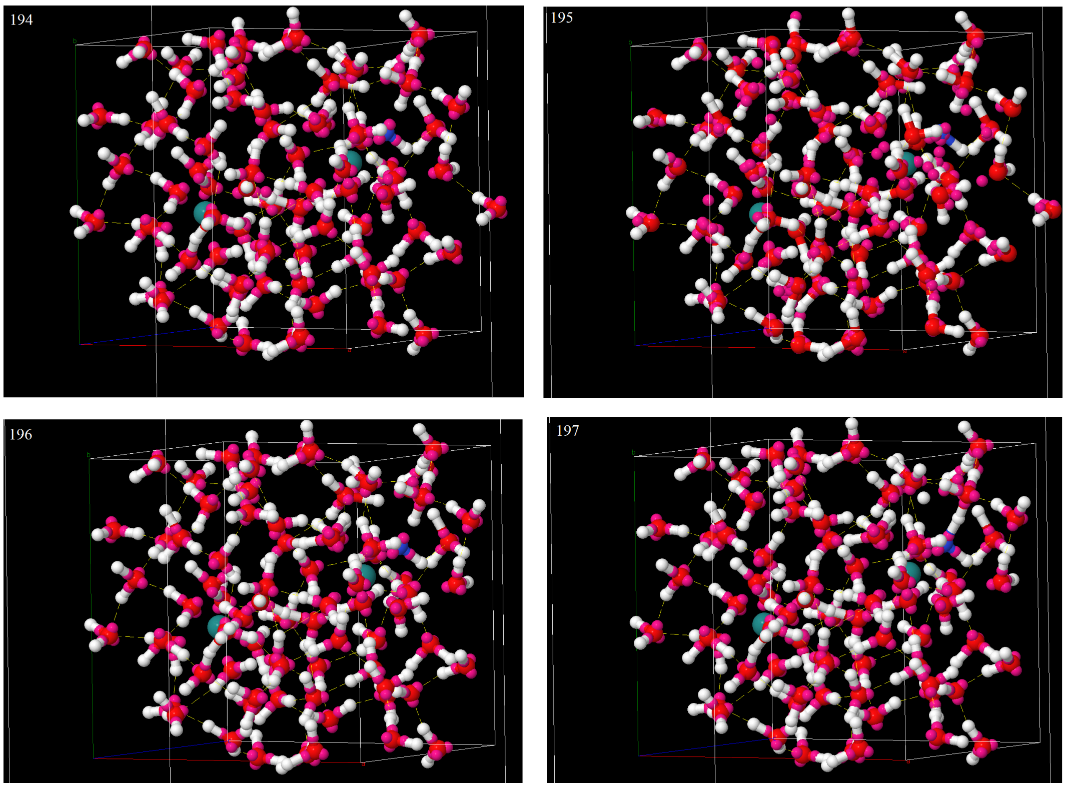

Figure 3.12: Frame-by-frame screenshots of trajectory 17 in the [6^+] ensemble visualized in Jmol from frame 194 to 197. Ruthenium is shown as green, oxygen as red, hydrogen as white, electrons as pink and the oxygen in OH^- as blue. Several lone Wannier centers can be observed in frame 195. The position of the Wannier centers also change drastically from frame 194 to 195 and frame 195 to 196, compared to their change from 196 to 197.

As seen in figure 3.12, the electrons rapidly change configuration from 194 to 195 and 195 to 196. Another notable observation is the lone hydrogen in frame 197. These frames indicate that the configuration in frame 195 is unphysical, and is caused by the SCF not converging.

It is therefore important to study the trajectories generated along with data such as the potential energy and visualization tools. As shown in figure 3.13, the movement of the OP seems smooth and physical, however the potential energy plot show that the path starts out with a phase point where the electronic configuration did not converge.

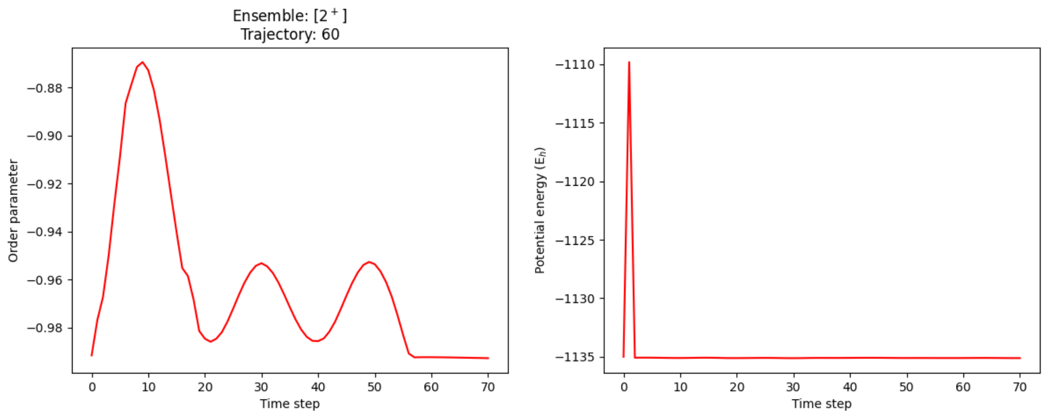

Figure 3.13: Plotted order parameter as a function of time (left) and potential energy (E_h) measured in Hartree, as a function of time (right) for trajectory 60 in the [2^+] ensemble. It is noticeable that there is an energy peak at the start of the simulation, however it is not clear from just looking at the order parameter plot. This is to demonstrate how smooth movement does not equate to the SCF converging for every phase point.

In contrast, there are no significant increases in the energy for trajectory 104 in the [5^+] ensemble as seen in figure 3.14. However, the order parameter shows uneven movement in frame 49. Potential energy is therefore not the only factor that should be used to determine whether a path is valid or not. As seen from our results, the water structure should also be taken into account to decide if a path is unphysical.

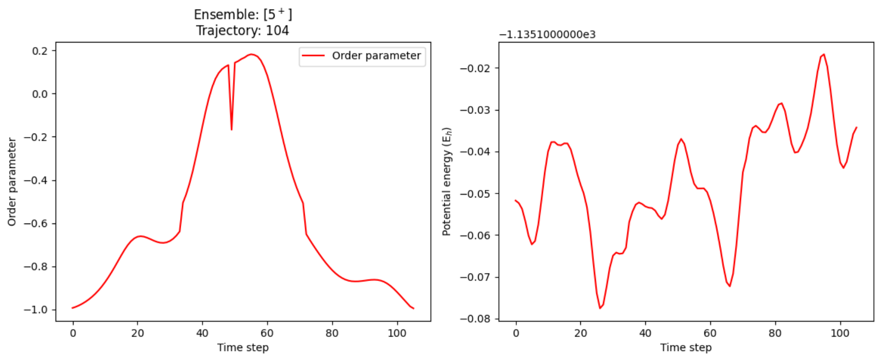

Figure 3.14: Plotted order parameter as a function of time (left) and potential energy measured in Hartree (E_h) as a function of time (right) for trajectory 104 in the [5^+] ensemble. The significant change in the order parameter at frame 49 does not correspond to any significant changes in the potential energy. This indicates that observing the potential energy is not enough to determine if a trajectory is unphysical.

We have attempted to improve on future simulations by either recalculating the SCF when it did not converge for a phase point, or rejecting paths containing one (or more) phase point(s), where the SCF did not converge. So far this has proven efficient and accurate as seen in figure 3.15. It is still possible that we obtain trajectories with uneven movements in the order parameter, however, this requires more testing.

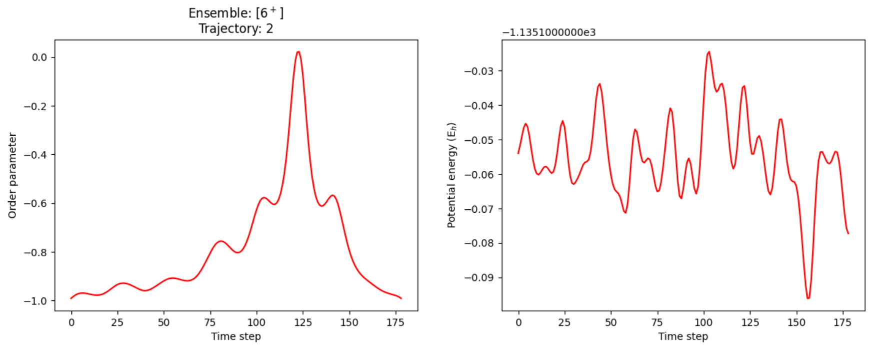

Figure 3.15: Trajectory 2 from the [6$^+$] ensemble in a simulation ran to test the recalculation and rejection based on energy convergence. Order parameter plotted as a function of time (left) and potential energy measured in Hartree (E_h) plotted as a function of time (right). This is a trajectory from a test simulation ran with the criterion in development where the SCF is recalculated, or the path is rejected.

3.2.3 Proton transfer

An important mechanism observed through the simulations was the position of the OH^- molecule. Because of the Grotthuss mechanism described earlier, the chain proton transfer reaction could occur to balance out other charge movement. To study this, we implemented an algorithm which iterated through every hydrogen to create a bond with the closest oxygen. For every snapshot, the distance between the oxygen (in the OH^-) and the ruthenium ions were calculated in the same way as the order parameter in equation (2.1).

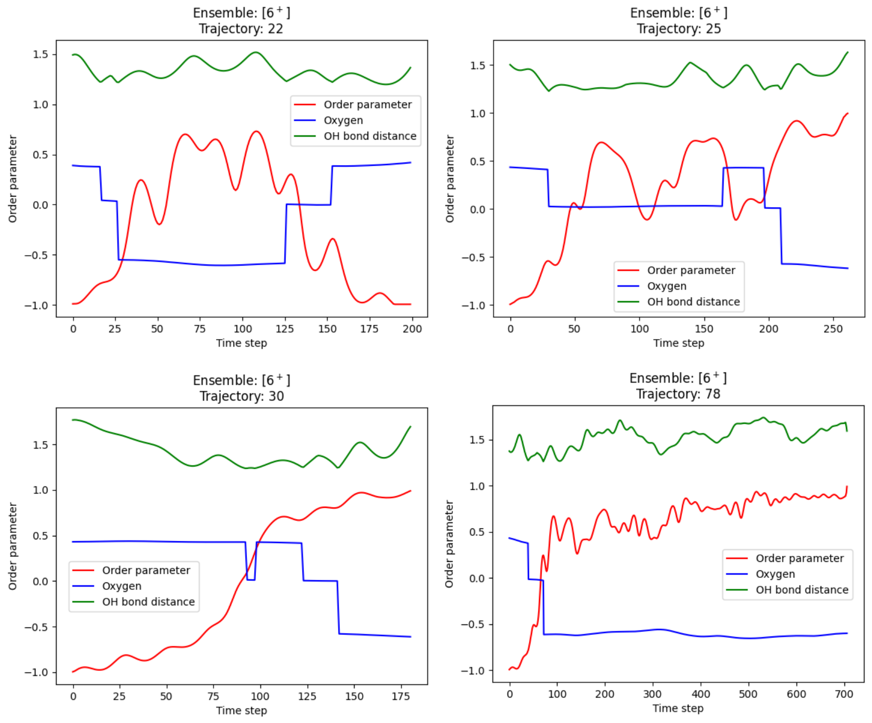

Figure 3.16: Four plots of the trajectories 22, 25, 30 and 78 in the [6^+] ensemble. The Wannier center, the distance between the oxygen in the OH^- molecule and the ruthenium ions with respect to equation (2.1), and the OH bond distance in OH^- is shown in red, blue and green respectively. These plots illustrate the relation between the movement of the OP and the location of the OH^- molecule as well as the OH bond length.

From these plots it was apparent that there was correlated movement between the OP and the “proton jumps” from the Grotthuss mechanism, as well as the OH bond length in the OH^- molecule. For every trajectory where the OP moved far enough away from state A, the proton transfer from an H_2O molecule to the OH^- occurred in the same direction. This resulted the OH^- molecule being closer to state A, when the OP moved towards state B and vice versa as seen in figure 3.16.

This is also clearly illustrated for trajectory 62 in the [6^+] ensemble in figure 3.17. The order parameter is highlighted as yellow for visualization purposes. As seen in the plot, the location of the OH^- molecule is initially closest to state B, however, at the end of the reaction it is closest to state A. The proton transfer mechanism which allows the OH^- to change its position by donating a proton to the previous OH^- is illustrated through these frames. It is also apparent how it moves in accordance to the movement of the order parameter.

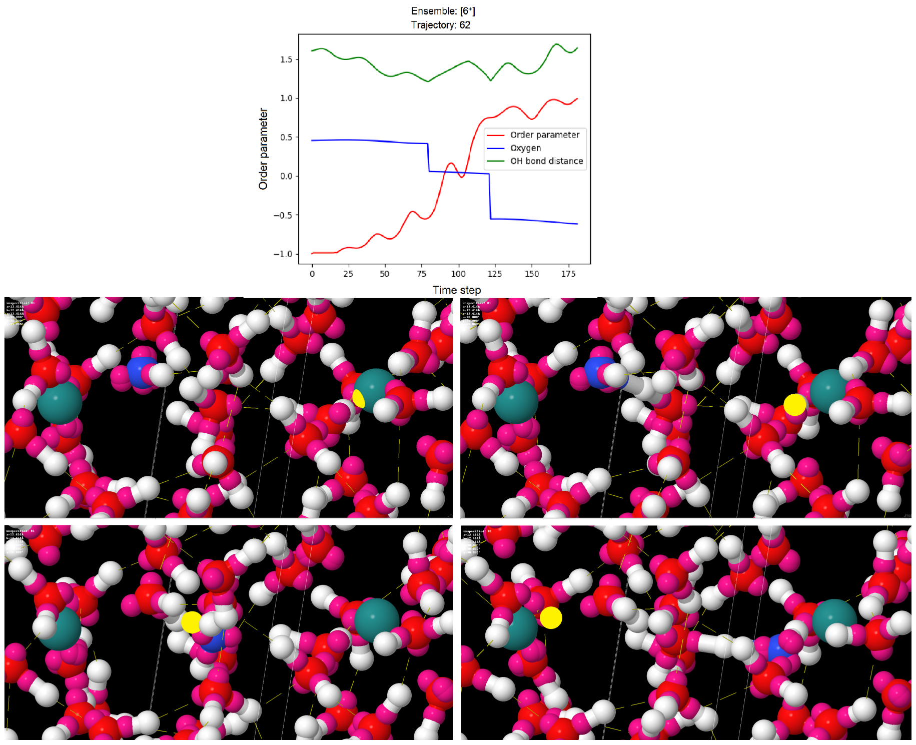

Figure 3.17: Order parameter is shown as a function of time (red), the oxygen position relative to the ruthenium ions in the OH^- (blue) and the OH bond length in the OH^- is shown for trajectory 62 in the $[6^+]$ ensemble. The Jmol illustration is presented under where the ruthenium are colored green, the electrons are pink, the hydrogens are white, the oxygens are red, except for the oxygen in the OH^- molecule, which is blue, and the order parameter is highlighted as yellow. These snapshots are from the frames 34, 45, 85 and 125, and show how the proton transfer mechanism occurs in accordance with the electron transfer reaction.

From figure 3.16, we notice how the OH bond length starts at 1.5 Å, and then contracts for all cases. When the hydrogen is donated, and a new oxygen becomes the OH^-, the bond length elongates afterwards as well. This gives clear correlation between these movements, however it is difficult to determine which effect initiates the other. From our water criteria we know that the spatial alignment is relevant to the self-exchange, but we have not been able to give a clear answer in how it happens. As seen in any of the plots in figure 3.16, the location of the OH^- does not change at specific values of the OP, but it always moves in the opposite direction of the OP. From our data the correlation between the OH^-, OH bond length and the order parameter is not clear, however it seems that there is a correlation present in a reactive trajectory.

3.3 Alternative softwares

Because the aim of the thesis is to accurately study self-exchange reaction between a pair of Ru(II) and Ru(III) in an aqueous solution, it is important to investigate how this should be done. The notable choices made in our simulations are the usage of PyRETIS and usage of the CP2K engine. In this section other possible approaches will be discussed, as well as the reasoning for our choices.

3.3.1 Physics software package

The chosen physics software package for our simulations was the CP2K engine. Because we want to run AIMD simulations we need a physics software package which supports this. Physics packages such as GROMACS[71] or LAMMPS[72] uses classical MD with interatomic potentials. As elaborated on in the theory section, it could be possible to run these simulations by using force fields such as ReaxFF, however there are two main issues. Because force fields do not model the electron, there is no way to describe this unless more development is made for force fields such as eReaxFF. The other issue is to train it with enough data. Because empirical force fields are trained using QM methods such as DFT or coupled cluster for the electronic structure, this would require a lot of computer resources.

A possible alternative to CP2K is VASP,[73] which is another simulation package. VASP is a commercial product, while CP2K is open source. CP2K is also the fastest package for AIMD calculations, albeit less accurate.[74] The ability to change the CP2K engine to suit our needs as well as being faster, is why CP2K was chosen for this project.

3.3.2 Path sampling

It would be possible to use other path sampling algorithms or libraries to run these simulations. There are not too many open source available path sampling methods, the two most notable choices for path sampling libraries are PyRETIS and OpenPathSampling (OPS).[75,76] These packages have different MC moves and different approaches to path sampling. PyRETIS uses more advanced and efficient moves such as the wire fencing move introduced in the theory section. We chose PyRETIS because of these moves, as well as the library is being developed by the research group in the institute of chemistry at NTNU.

4 Future work

A core part of this thesis was the development of a simulation setup to study the self-exchange between Ru(II) and Ru(III). This involved studying the system to improve the sampling algorithms of the reaction. Placement of the interfaces, which properties to study and the implemented rejection criteria allows for continuation from where this project ended.

4.1 Studying self-exchange mechanism

To understand the mechanisms of the reaction it is important to know what should be investigated. A possible explanation to why the self-exchange occurs is the electric field in the system. Depending on the alignment of the water molecules, the electric field affecting the electron at the Ru(II) changes.[70,77] To further investigate the mechanisms of the reaction, it would be possible to run these simulations and calculate the electric field at every time frame. By using Coulomb's law the electrostatic force can be expressed as

$$\mathbf{F} = k_e \frac{q_1 \cdot q_2}{r^2} \tag{4.1}$$

where k_e is Coulomb's constant, q_1 and q_2 are the charges of the interacting particles and r is the vectorial distance between the particles.[78] By calculating the electrostatic force affecting the electron from every particle, a total vector field should then indicate if the self-exchange reaction is due to the spatial alignment of the water molecules.

Another important property is the ionization energy for the ruthenium ions. Because CP2K calculates the orbital energies for the system, it is possible to track these frame by frame throughout the reaction. By studying the energy required to remove an electron from Ru(II) and the energy required to insert an electron into Ru(III), it is possible to determine when the electron transfer starts. One would expect that the energy required to remove an electron for Ru(II) goes down until it is lower than the energy required to insert an electron to Ru(III). By knowing when the electron is removed from Ru(II), it is easier to determine which properties that dictates the reaction. It would also aid in studying the causes for unreactive trajectories.

4.2 Improved algorithms

As PyRETIS is under development the MC moves will continue to improve, which will result in faster and more accurate study of the system. From this project we have uncovered the effect of badly converged phase points. As an improvement, we chose to recalculate these phase points, or reject the path if the configuration does not converge. This implementation should reject unphysical trajectories as early as possible, increasing the efficiency and accuracy of the simulations.

As shown in the results there are trajectories with no significant energy increases, which still contains discontinuous movement of the Wannier center. This would indicate that there are more factors than the energy in the system which should be studied. From the discussion, the OH-bond in the OH^- molecule fluctuate between 1.2 and 1.5 Å. For some of the unphysical trajectories, we noticed that the OH bond distance was up to 1.9 Å. It could then be possible to implement a rejection criterion based on too high or too low OH-bond distances. This would require more data to implement, but highlights that it is worth investigating more properties to detect and reject unphysical configurations.

5 Conclusion

The aim of this thesis was to study the self-exchange reaction of Ru(II) and Ru(III) in an aqueous solution using ab initio molecular dynamics simulation and replica exchange transition interface sampling from the PyRETIS library. Through four iterations of PyRETIS simulations, the interfaces for the RETIS simulations were placed at [-0.990, -0.970, -0.960, -0.700, -0.500, 0.000, 0.990]. Due to the technical difficulties related to the highly decorrelated electronic structure when the electron moved between the ruthenium ions, most of the time was used to overcome these difficulties. Two rejection criteria were implemented, one where a path was rejected if it contained more than six water complexes at any phase point, the other rejected a path if it had more than two water complexes for more than 150 MD steps. A third criterion was implemented to avoid accepting unstable configurations, which was done by continuing the simulation as long as there were any water complexes present.

The results from these simulations are therefore based on very few MC steps. From 114 TIS cycles the rate constant was calculated to be $3.121 \cdot 10^{-4}$ [1/fs] with a relative error of 129.7%. The simulations of the self-exchange was studied revealing a correlation between the movement of the Wannier center with the movement of the OH^- molecule, as well as the OH bond length in the OH^- molecule. These simulations are now continued by the research group of T. S. van Erp, where they are implementing a method that converges electronic structures where the SCF did not converge.

References

[1] J. Strümpfer, M. Şener, and K. Schulten, "How quantum coherence assists photosynthetic light-harvesting," *The Journal of Physical Chemistry Letters*, vol. 3, no. 4, pp. 536–542, 2012. PMID: 22844553.

[2] S. Weber, "Light-driven enzymatic catalysis of dna repair: a review of recent biophysical studies on photolyase," *Biochimica et Biophysica Acta (BBA) - Bioenergetics*, vol. 1707, no. 1, pp. 1–23, 2005. Functional Redox Radicals in Proteins.

[3] A. Houmam, "Electron transfer initiated reactions: Bond formation and bond dissociation," *Chemical Reviews*, vol. 108, no. 7, pp. 2180–2237, 2008. PMID: 18620366.

[4] L. Wu, X. Guo, J. Wang, Q. Guo, Z. Liu, and Y. Liu, "Kinetic studies on the single electron transfer reaction between 2, 2, 6, 6-tetramethylpiperidine oxoammonium ions and phenothiazines: the application of marcus theory," *Science in China Series B: Chemistry*, vol. 42, pp. 138–144, 1999. 10.1007/BF02875509.

[5] R. Iftimie, P. Minary, and M. E. Tuckerman, "Ab initio molecular dynamics: Concepts, recent developments, and future trends," *Proceedings of the National Academy of Sciences*, vol. 102, no. 19, pp. 6654–6659, 2005.

[6] C. Dellago, P. G. Bolhuis, and P. L. Geissler, *Transition Path Sampling*. Springer International Publishing, 2001.

[7] T. S. van Erp, D. Moroni, and P. G. Bolhuis, "A novel path sampling method for the calculation of rate constants," *The Journal of Chemical Physics*, vol. 118, no. 17, pp. 7762–7774, 2003.

[8] R. Cabriolu, K. M. Skjelbred Refsnes, P. G. Bolhuis, and T. S. van Erp, "Foundations and latest advances in replica exchange transition interface sampling," *The Journal of Chemical Physics*, vol. 147, no. 15, p. 152722, 2017.

[9] A. Lervik, E. Riccardi, and T. S. van Erp, "Pyretis: A well-done, medium-sized python library for rare events," *Journal of Computational Chemistry*, vol. 38, no. 28, pp. 2439–2451, 2017.

[10] E. Riccardi, A. Lervik, S. Roet, O. Aarøen, and T. S. van Erp, "Pyretis 2: An improbability drive for rare events," *Journal of Computational Chemistry*, vol. 41, no. 4, pp. 370–377, 2020.

[11] D. T. Zhang, E. Riccardi, and T. S. van Erp, "Path sampling with sub-trajectory moves." In preparation (2022).

[12] J. B. Clarke, J. W. Hastie, L. H. E. Kihlborg, R. Metselaar, and M. M. Thackeray, "Definitions of terms relating to phase transitions of the solid state (iupac recommendations 1994)," *Pure and Applied Chemistry*, vol. 66, no. 3, pp. 577–594, 1994.

[13] C. de Grotthuss, "Sur la décomposition de l'eau et des corps qu'elle tient en dissolution à l'aide de l'électricité galvanique," *Ann. Chim*, vol. 58, p. 54–73, 1806.

[14] M. Chen, L. Zheng, B. Santra, H.-Y. Ko, R. A. DiStasio Jr, M. L. Klein, R. Car, and X. Wu, "Hydroxide diffuses slower than hydronium in water because its solvated structure inhibits correlated proton transfer," *Nature Chemistry*, vol. 10, 3 2018.

[15] B. Peters, "Chapter 20 - reaction coordinates and mechanisms," in *Reaction Rate Theory and Rare Events Simulations* (B. Peters, ed.), pp. 539–571, Amsterdam: Elsevier, 2017.

[16] R. A. Marcus, "Electron transfer reactions in chemistry: Theory and experiment (nobel lecture)," *Angewandte Chemie International Edition in English*, vol. 32, no. 8, pp. 1111–1121, 1993.

[17] A. D. M. Jr., B. Brooks, I. C. L. Brooks, L. Nilsson, B. Roux, Y. Won, and M. Karplus, *The Encyclopedia of Computational Chemistry*, vol. 1. R. Schleyer et al., 1998.

[18] P. A. M. Dirac, *The Principles of Quantum Mechanics*. Clarendon Press, 1930.

[19] M. Born and R. Oppenheimer, "Zur quantentheorie der molekeln," *Annalen der Physik*, vol. 389, no. 20, pp. 457–484, 1927.

[20] P. Atkins and R. Friedman, *Molecular Quantum Mechanics*. OUP Oxford, 2011.

[21] U. Ryde, "Chapter six - qm/mm calculations on proteins," in *Computational Approaches for Studying Enzyme Mechanism Part A* (G. A. Voth, ed.), vol. 577 of *Methods in Enzymology*, pp. 119–158, Academic Press, 2016.

[22] E. Rossi, G. L. Bendazzoli, S. Evangelisti, and D. Maynau, "A full-configuration benchmark for the n2 molecule," *Chemical Physics Letters*, vol. 310, no. 5, pp. 530–536, 1999.

[23] C. Riplinger, B. Sandhoefer, A. Hansen, and F. Neese, "Natural triple excitations in local coupled cluster calculations with pair natural orbitals," *The Journal of Chemical Physics*, vol. 139, no. 13, p. 134101, 2013.

[24] A. R. Leach, *Molecular Modelling Principles and Applications.* Pearson Education Limited, 2001.

[25] P. M. Morse, "Diatomic molecules according to the wave mechanics. ii. vibrational levels," *Phys. Rev.*, vol. 34, pp. 57–64, Jul 1929.

[26] R. A. Buckingham and J. E. Lennard-Jones, "The classical equation of state of gaseous helium, neon and argon," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 168, no. 933, pp. 264–283, 1938.

[27] B. R. Brooks, C. L. Brooks III, A. D. Mackerell Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus, "Charmm: The biomolecular simulation program," *Journal of Computational Chemistry*, vol. 30, no. 10, pp. 1545–1614, 2009.

[28] W. L. Jorgensen and J. Tirado-Rives, "The opls [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin," *Journal of the American Chemical Society*, vol. 110, no. 6, pp. 1657–1666, 1988. PMID: 27557051.

[29] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case, "Development and testing of a general amber force field," *Journal of computational chemistry*, vol. 25, p. 1157—1174, July 2004.

[30] A. Warshel and R. M. Weiss, "An empirical valence bond approach for comparing reactions in solutions and in enzymes," *Journal of the American Chemical Society*, vol. 102, no. 20, pp. 6218–6226, 1980.

[31] T. Senftle, S. Hong, M. Islam, S. Kylasa, Y. Zheng, Y. Shin, C. Junkermeier, R. Engel-Herbert, M. Janik, H. Aktulga, T. Verstraelen, A. Grama, and A. Van Duin, "The reaxff reactive force-field: Development, applications and future directions," *npj Computational Materials*, vol. 2, Mar. 2016.

[32] M. Moqadam, E. Riccardi, T. T. Trinh, P.-O. Åstrand, and T. S. van Erp, "A test on reactive force fields for the study of silica dimerization reactions," *The Journal of Chemical Physics*, vol. 143, no. 18, p. 184113, 2015.

[33] M. M. Islam, G. Kolesov, T. Verstraelen, E. Kaxiras, and A. C. T. van Duin, "ereaxff: A pseudoclassical treatment of explicit electrons within reactive force field simulations," *Journal of Chemical Theory and Computation*, vol. 12, no. 8, pp. 3463–3472, 2016. PMID: 27399177.

[34] S. Ekesan, S. Kale, and J. Herzfeld, "Transferable pseudoclassical electrons for aufbau of atomic ions," *Journal of Computational Chemistry*, vol. 35, no. 15, pp. 1159–1164.

[35] Shchygol, Ganna and Yakovlev, Alexei and Trnka, Tomáš and van Duin, Adri CT and Verstraelen, Toon, "ReaxFF parameter optimization with Monte-Carlo and evolutionary algorithms : guidelines and insights," *JOURNAL OF CHEMICAL THEORY AND COMPUTATION*, vol. 15, no. 12, pp. 6799–6812, 2019.

[36] J. Pan, "Scaling up system size in materials simulation," *Nature Computational Science*, vol. 1, 2021.

[37] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, pp. B864–B871, Nov 1964.

[38] A. D. Becke, "Perspective: Fifty years of density-functional theory in chemical physics," *The Journal of Chemical Physics*, vol. 140, no. 18, p. 18A301, 2014.

[39] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," *Phys. Rev.*, vol. 140, pp. A1133–A1138, Nov 1965.

[40] J. P. Perdew and A. Zunger, "Self-interaction correction to density-functional approximations for many-electron systems," *Phys. Rev. B*, vol. 23, pp. 5048–5079, May 1981.

[41] J. P. Perdew and Y. Wang, "Accurate and simple analytic representation of the electron-gas correlation energy," *Phys. Rev. B*, vol. 45, pp. 13244–13249, Jun 1992.

[42] S. H. Vosko, L. Wilk, and M. Nusair, "Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis," *Canadian Journal of Physics*, vol. 58, no. 8, pp. 1200–1211, 1980.

[43] A. D. Becke, "Density-functional exchange-energy approximation with correct asymptotic behavior," *Phys. Rev. A*, vol. 38, pp. 3098–3100, Sep 1988.

[44] C. Lee, W. Yang, and R. G. Parr, "Development of the colle-salvetti correlation-energy formula into a functional of the electron density," *Phys. Rev. B*, vol. 37, pp. 785–789, Jan 1988.

[45] C. J. Cramer, *Essentials of Computational Chemistry: Theories and Models.* John Wiley and Sons, 2004.

[46] C. Lee and C. Sosa, "Local density component of the lee–yang–parr correlation energy functional," *The Journal of Chemical Physics*, vol. 100, no. 12, pp. 9018–9024, 1994.

[47] J. Jung, C. Kobayashi, K. Kasahara, C. Tan, A. Kuroda, K. Minami, S. Ishiduki, T. Nishiki, H. Inoue, Y. Ishikawa, M. Feig, and Y. Sugita, "New parallel computing algorithm of molecular dynamics for extremely huge scale biological systems," *Journal of Computational Chemistry*, vol. 42, no. 4, pp. 231–241, 2021.

[48] S. Sharma, P. Kumar, and R. Chandra, "Chapter 2 - overview of biovia materials studio, lammps, and gromacs," in *Molecular Dynamics Simulation of Nanocomposites Using BIOVIA Materials Studio, Lammps and Gromacs* (S. Sharma, ed.), Micro and Nano Technologies, pp. 39–100, Elsevier, 2019.

[49] P. Gkeka, *Molecular dynamics studies of peptide-membrane interactions: insights from coarse-grained models.* PhD thesis, 01 2010.

[50] E. Paquet and H. L. Viktor, "Computational methods for ab initio molecular dynamics," 2018.

[51] R. Car and M. Parrinello, "Unified approach for molecular dynamics and density-functional theory," *Phys. Rev. Lett.*, vol. 55, pp. 2471–2474, Nov 1985.

[52] R. Iftimie and M. E. Tuckerman, "Decomposing total ir spectra of aqueous systems into solute and solvent contributions: A computational approach using maximally localized wannier orbitals," *The Journal of Chemical Physics*, vol. 122, no. 21, p. 214508, 2005.

[53] G. H. Wannier, "The structure of electronic excitation levels in insulating crystals," *Phys. Rev.*, vol. 52, pp. 191–197, Aug 1937.

[54] P. Fraundorf, "Heat capacity in bits," *American Journal of Physics*, vol. 71, no. 11, pp. 1142–1151, 2003.

[55] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 04 1970.

[56] A. A. Markov, "Extension of the limit theorems of probability theory to a sum of variables connected in a chain," *Dynamic probabilistic systems*, vol. 1, pp. 552–577, 1971.

[57] C. C. Moore, "Ergodic theorem, ergodic theory, and statistical mechanics," *Proceedings of the National Academy of Sciences*, vol. 112, no. 7, pp. 1907–1911, 2015.

[58] W. H. McCrea, "The principles of statistical mechanics. by r. c. tolman. pp. xix, 661. 40s. 1938. international series of monographs on physics. (oxford)," *The Mathematical Gazette*, vol. 23, no. 256, 1939.

[59] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, "Transition path sampling: Throwing ropes over rough mountain passes, in the dark," *Annual Review of Physical Chemistry*, vol. 53, no. 1, pp. 291–318, 2002. PMID: 11972010.

[60] C. Dellago, P. Bolhuis, and P. Geissler, *Transition Path Sampling Methods*, pp. 349–391. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[61] T. S. Van Erp, *Dynamical Rare Event Simulation Techniques for Equilibrium and Nonequilibrium Systems*, ch. 2, pp. 27–60. John Wiley & Sons, Ltd, 2012.

[62] T. S. van Erp and P. G. Bolhuis, "Elaborating transition interface sampling methods," *Journal of Computational Physics*, vol. 205, no. 1, pp. 157–181, 2005.

[63] E. Riccardi, O. Dahlen, and T. S. van Erp, "Fast decorrelating monte carlo moves for efficient path sampling," *The Journal of Physical Chemistry Letters*, vol. 8, no. 18, pp. 4456–4460, 2017. PMID: 28857565.

[64] T. van Erp, "Reaction rate calculation by parallel path swapping," *Physical review letters*, vol. 98, p. 268301, 07 2007.

[65] M. Själander, M. Jahre, G. Tufte, and N. Reissmann, "EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure," 2019.

[66] J. S. Hub, B. L. de Groot, H. Grubmüller, and G. Groenhof, "Quantifying artifacts in ewald simulations of inhomogeneous systems with a net charge," *Journal of Chemical Theory and Computation*, vol. 10, no. 1, pp. 381–390, 2014. PMID: 26579917.

[67] T. D. Kühne, M. Iannuzzi, M. Del Ben, V. V. Rybkin, P. Seewald, F. Stein, T. Laino, R. Z. Khaliullin, O. Schütt, F. Schiffmann, D. Golze, J. Wilhelm, S. Chulkov, M. H. Bani-Hashemian, V. Weber, U. Borštnik, M. Taillefumier, A. S. Jakobovits, A. Lazzaro, H. Pabst, T. Müller, R. Schade, M. Guidon, S. Andermatt, N. Holmberg, G. K. Schenter, A. Hehn, A. Bussy, F. Belleflamme, G. Tabacchi, A. Glöß, M. Lass, I. Bethune, C. J. Mundy, C. Plessl, M. Watkins, J. VandeVondele, M. Krack, and J. Hutter, "Cp2k: An electronic structure and molecular dynamics software package - quickstep: Efficient and accurate electronic structure calculations," *The Journal of Chemical Physics*, vol. 152, no. 19, p. 194103, 2020.

[68] A. Tiwari and B. Ensing, "Reactive trajectories of the ru2+/3+ self-exchange reaction and the connection to marcus' theory," *Faraday Discuss.*, vol. 195, pp. 291–310, 2016.

[69] Jmol development team, "Jmol," *http://jmol.sourceforge.net/*, 2016.

[70] T. S. van Erp, "Solvent effects on chemistry with alcohols. an ab initio study," 2003.

[71] P. Bauer, B. Hess, and E. Lindahl, "Gromacs 2022 manual," Feb. 2022.

[72] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Comp. Phys. Comm.*, vol. 271, p. 108171, 2022.

[73] J. Hafner and G. Kresse, *The Vienna AB-Initio Simulation Program VASP: An Efficient and Versatile Tool for Studying the Structural, Dynamic, and Electronic Properties of Materials*, pp. 69–82. Boston, MA: Springer US, 1997.

[74] D. Yokelson, N. V. Tkachenko, R. Robey, Y. W. Li, and P. A. Dub, "Performance analysis of cp2k code for ab initio molecular dynamics on cpus and gpus," *Journal of Chemical Information and Modeling*, vol. 62, no. 10, pp. 2378–2386, 2022. PMID: 35451847.

[75] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, "Openpathsampling: A python framework for path sampling simulations. 2. building and customizing path ensembles and sample schemes," *Journal of Chemical Theory and Computation*, vol. 15, no. 2, pp. 837–856, 2019. PMID: 30359525.

[76] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, "Openpathsampling: A python framework for path sampling simulations. 1. basics," *Journal of Chemical Theory and Computation*, vol. 15, no. 2, pp. 813–836, 2019. PMID: 30336030.

[77] S. D. Fried, L.-P. Wang, S. G. Boxer, P. Ren, and V. S. Pande, "Calculations of the electric fields in liquid solutions," *The Journal of Physical Chemistry B*, vol. 117, no. 50, pp. 16236–16248, 2013. PMID: 24304155.

[78] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics.* Wiley, 2018.

Appendix A: Code for the order parameter

```python
# -*- coding: utf-8 -*-
# Copyright (c) 2021, PyRETIS Development Team.
# Distributed under the LGPLv2.1+ License. See LICENSE for more
   info.
"""The order parameter for the hysteresis example."""
from copy import *
import logging
import glob
import numpy as np
import MDAnalysis.coordinates.XYZ as md2
from MDAnalysis.analysis import distances
import os, os.path
from pyretis.orderparameter.orderparameter import OrderParameter
from pyretis.inout.formats.xyz import (
write_xyz_trajectory,
read_xyz_file
)
from pyretis.engines.cp2k import (
write_for_step_vel,
write_for_continue,
CP2KEngine
)
from pyretis.inout.settings import parse_settings_file

logger = logging.getLogger(__name__)  # pylint: disable=invalid-
   name
logger.addHandler(logging.NullHandler())
vpotList = []

class OrderX(OrderParameter):
    """A positional order parameter.

    Order parameter for the hysteresis example. In addition to
   using
    the position, we also use the energy to tell if we are in
   states A/B.


    Attributes
    ----------
    index : integer
        This is the index of the atom which will be used, i.e.
        system.particles.pos[index] will be used.
    inter_a : float
        An interface such that we are in state A for postions <
   inter_a.
```

```
    inter_b : float
        An interface such that we are in state B for postions > inter_b.
    energy_a : float
        An energy such that we are in state A for potential energy < energy_a.
    energy_b : float
        An energy such that we are in state A for potential energy < energy_b.
    dim : integer
        This is the dimension of the coordinate to use.
        0, 1 or 2 for 'x', 'y' or 'z'.
    periodic : boolean
        This determines if periodic boundaries should be applied to
        the position or not.

    """

    def __init__(self, index,  periodic=False):
        """Initialise the order parameter.

        Parameters
        ----------
        index : tuple of ints
            This is the indices of the atoms we will use the position of.
        periodic : boolean, optional
            This determines if periodic boundary conditions should be
            applied to the position.

        """
        pbc = 'Periodic' if periodic else 'Non-periodic'
        txt = '{} distance, particles {} and {}'.format(
            pbc,
            index[0],
            index[1]
        )
        super().__init__(description=txt, velocity=False)
        self.periodic = periodic
        self.index = index

    def calculate(self, system):
        """Calculate the order parameter.

        Here, the order parameter is just the distance between two
        particles.

        Parameters
        ----------
        system : object like :py:class:`.System`
            The object containing the positions and box used for The
            calculation.

        Returns
        -------
        out : list of floats
            The rate-of-change of the distance order parameter.
        """
        box = [12.4138, 12.4138, 12.4138, 90.0, 90.0, 90.0]
        at, o, ru = 193, 64, 2  # no. of atoms

        string = os.path.basename(os.path.normpath(system.particles.config[0]))[:-4]
        s2 = system.particles.config[0][0:-(len(string)+4)]
        list_of_files = glob.glob(s2 + '*')
        particles = system.particles
        p1 = particles.pos[self.index[0]]
        p2 = particles.pos[self.index[1]]
        key = False
        eng_set = parse_settings_file('./retis.rst')['engine']
        eng = CP2KEngine(eng_set['cp2k'],
                            eng_set['input_path'],
                            eng_set['timestep'],
                            eng_set['subcycles'],
                            eng_set['extra_files'])
        eng.exe_dir = os.path.dirname(system.particles.config[0])

        HOMOs = [s for s in list_of_files if 'HOMO' in s]
        if HOMOs != []:
            f1, f2, latest_s1, latest_s2, key = picker(HOMOs, p1, p2, edge=False)
            if not key:
                f1, f2, latest_s1, latest_s2, key = picker(HOMOs, p1, p2, edge=True)
            if key:
                traj1 = md2.XYZReader(latest_s1, box=box)
                traj2 = md2.XYZReader(latest_s2, box=box)
        if not key:
            traj1, traj2, f1, f2, key, out_files = engine(eng, 'wann', system, particles, s2, p1, p2, box, 0, key)

        if not key:
            exit('failure, no HOMO files')

        dist3, OP, spin, loc, bad = finder(traj1, traj2, f1, f2,
```

```python
at, ru, o, box)

    # have to make a check to see if we did eng call or not..
    if bad:
        traj1, traj2, f1, f2, key, out_files = engine(eng, 'wann_0', system, particles, s2, p1, p2, box, 1, key, False)
        dist3, OP, spin, loc, bad = finder(traj1, traj2, f1, f2, at, ru, o, box)
        if bad:
            for i in range(2)[1:]:
                restart_file = os.path.join(eng.exe_dir, out_files['restart'])
                prestart_file = os.path.join(eng.exe_dir, 'previous.restart')
                wave_file = os.path.join(eng.exe_dir, out_files['wfn'])
                pwave_file = os.path.join(eng.exe_dir, 'previous.wfn')
                eng._movefile(restart_file, prestart_file)
                eng._movefile(wave_file, pwave_file)
                traj1, traj2, f1, f2, key, out_files = engine(eng, 'wann_'+str(i), system, particles, s2, p1, p2, box, 1, key, True)
                dist3, OP, spin, loc, bad = finder(traj1, traj2, f1, f2, at, ru, o, box)
                if not bad:
                    break
    if not bad:
        try:
            for _, files in out_files.items():
                eng._removefile(files)
            eng._remove_files(
                eng.exe_dir,
                eng._find_backup_files(eng.exe_dir))
        except:

        # Calculate state here. Use whichever traj1
        # The frame is called f1 (the current frame), traj1[f1][ATOM]
        # Add information about the water cluster

        OPfake = OP
        OPtrue = OP
        complexCount = 0

        vpotList.append(system.particles.vpot)

        if(OPtrue > 0.990):
            stable, complexCount = checkStable(traj1)
```

```
                if(stable == False):
                    OPfake = 0.989
            elif(OPtrue < -0.990):
                stable, complexCount = checkStable(traj1)
                if (stable == False):
                    OPfake = -0.989
            return [OPfake, OPtrue, complexCount, dist3[0][0]]
        else:
            return [9000]

def checkStable(trajectory):
    traj_s1 = trajectory
    L = 12.4138
    dH = 0.91 / 2.
    dO = 1.81 / 2.
    cutoffs = {"H": dH, "O": dO}

    def DISTANCE(c1, c2, L):
        vector = c1 - c2
        if L is not None: vector -= L * np.around(vector / L)
        d = np.sqrt(sum(vector * vector))
        return d

    def CONNECTED(at1, at2, cutoffs, L):
        c1, el1 = at1[0], at1[1]
        cutoff1 = cutoffs[el1]
        c2, el2 = at2[0], at2[1]
        cutoff2 = cutoffs[el2]
        d = DISTANCE(c1, c2, L)
        return d < cutoff1 + cutoff2

    def EXTRACTNEIGHBORSFROMLIST(atom, leftover, cutoffs, L):
        indexleftover = 0
        extract = []
        while indexleftover < len(leftover):
            secatom = leftover[indexleftover]
            if CONNECTED(atom, secatom, cutoffs, L):
                extract += [secatom]
                del leftover[indexleftover]
            else:
                indexleftover += 1
        return extract, leftover

    def MOLECLIST(atomlist, L, cutoffs):
        moleclist = []
        leftover = deepcopy(atomlist)
        while len(leftover) > 0:
            mol = []
            mol += [leftover[0]]
            del leftover[0]
            iat = 0
            while iat < len(mol):
                atom = mol[iat]
                neighbors, leftover = EXTRACTNEIGHBORSFROMLIST(
    atom, leftover, cutoffs, L)
                mol += neighbors
                iat += 1
            moleclist += [mol]
        return moleclist

    atomList = []
    # Make atomList
    for i in range(2, 193):
        if (i < 66):
            atomList.append([traj_s1[len(traj_s1) - 1][i], "O"])
        else:
            atomList.append([traj_s1[len(traj_s1) - 1][i], "H"])

    myList = MOLECLIST(atomList, L, cutoffs)

    # Make molecules
    molecList = []
    #molecString = ""
    for i in myList:
        tempMolecule = ""
        for j in range(len(i)):
            tempMolecule += i[j][1]
            #molecString += i[j][1]
        #molecString += " | "
        molecList.append(tempMolecule)

    flag = True
    complexCount = 0
    for i in molecList:
        if (i != "OH" and i != "OHH"):
            flag = False
            complexCount += 1
    return flag, complexCount

def picker(HOMOs, p1, p2, edge=False):
    HOMO_s1 = [s for s in HOMOs if 'HOMO_centers_s1' in s]
    HOMO_s2 = [s for s in HOMOs if 'HOMO_centers_s2' in s]
    loop_key = False
    f1 = None
    f2 = None
    key = False
    if len(HOMO_s1) != len(HOMO_s2):
        exit('len homo_s1 != len homo_s2')
```

```python
    for k in range(len(HOMO_s1)):
        latest_s1 = min(HOMO_s1, key=os.path.getctime)
        latest_s2 = min(HOMO_s2, key=os.path.getctime)
        traj_s1 = md2.XYZReader(latest_s1)
        traj_s2 = md2.XYZReader(latest_s2)
        if not edge:   # Most accurate homos exist
            rev_s1 = list(range(len(traj_s1)))[1:-1]
            rev_s2 = list(range(len(traj_s2)))[1:-1]
        else:
            rev_s1 = [len(traj_s1)-1, 0]
            rev_s2 = [len(traj_s2)-1, 0]
        for i in rev_s1:
            if np.max(np.abs(np.float32(p1) - traj_s1[i][0])) < 10e-7 and \
                    np.max(np.abs(np.float32(p2) - traj_s1[i][1])) < 10e-7:
                f1 = i
                break
        for i in rev_s2:
            if np.max(np.abs(np.float32(p1) - traj_s2[i][0])) < 10e-7 and \
                    np.max(np.abs(np.float32(p2) - traj_s2[i][1])) < 10e-7:
                f2 = i
                loop_key = True
                key = True
                break
        if HOMO_s1 == [] or HOMO_s2 == [] or loop_key:
            break
        HOMO_s1.remove(latest_s1)
        HOMO_s2.remove(latest_s2)
    if key:
        return f1, f2, latest_s1, latest_s2, key
    else:
        return None, None, None, None, key


def engine(eng, name, system, particles, s2, p1, p2, box, cycle, key, cont=False):
    conf = next(read_xyz_file(system.particles.config[0]))
    input_config = os.path.join(eng.exe_dir, name +'.xyz')
    write_xyz_trajectory(input_config, particles.pos, particles.vel,
                         conf['atomname'], system.box.cell, append=False)
    run_file = os.path.join(eng.exe_dir, name + '.inp')
    if not cont:
        write_for_step_vel(eng.input_files['template'], run_file,
                         0, cycle, name+'.xyz',
                                system.particles.vel, name=name,
   print_freq=1)
    else:
        write_for_continue(eng.input_files['template'], run_file,
                                0, cycle, name=name)

    eng.add_input_files(eng.exe_dir)
    out_files = eng.run_cp2k(run_file, name)
    list_of_files = glob.glob(s2 + '*')
    HOMOs = [s for s in list_of_files if name + '-HOMO' in s]
    f1 = -1
    f2 = -1
    for k in range(len(HOMOs)):
        latest = max(HOMOs, key=os.path.getctime)
        traj1 = md2.XYZReader(latest)
        if np.max(np.abs(np.float32(p1) - traj1[f1][0])) < 10e-7
   and \
        np.max(np.abs(np.float32(p2) - traj1[f1][1])) < 10e-7 :
            traj2 = md2.XYZReader(latest, box=box)
            HOMOs.remove(latest)
            latest2 = max(HOMOs, key=os.path.getctime)
            traj1 = md2.XYZReader(latest2, box=box)
            key = True
            break
        HOMOs.remove(latest)
    if key:
        return traj1, traj2, f1, f2, key, out_files
    else:
        return None, None, None, None, key, None


def finder(traj1, traj2, f1, f2, at, ru, o, box):
    atom_array = [0] * len(traj1[f1][:ru+o])
    dic = dict.fromkeys(list(range(len(atom_array))))
    bad = False
    for i in dic:
        dic[i] = {'x': 0}
        dic[i]['dist'] = []
    # generate atom_array and dict
    for i in range(len(traj1[f1][at:][:, 0])):
        dist_arr = distances.distance_array(traj1[f1][at+i], traj1
   [f1][:ru+o], box=box)[0]
        loc = np.argmin(dist_arr)
        atom_array[loc] += 1
        dic[loc]['x'] += 1
        dic[loc]['dist'].append((i+at, dist_arr[loc], 's1', loc))
    for i in range(len(traj2[f2][at:][:, 0])):
        dist_arr = distances.distance_array(traj2[f2][at+i], traj2
   [f2][:ru+o], box=box)[0]
```

```python
     loc = np.argmin(dist_arr)
     atom_array[loc] += 1
     dic[loc]['x'] += 1
     dic[loc]['dist'].append((i+at, dist_arr[loc], 's2', loc))
 # assume that excess electron exist at Ru xyz[0]
 # calculate op, dist:
 if atom_array[0] == 6:
     loc = dic[0]['dist'][np.argmax([a[1] for a in dic[0]['dist']])][0]
     spin = dic[0]['dist'][np.argmax([a[1] for a in dic[0]['dist']])][2]
 elif (atom_array[0] == atom_array[1] == 5):
     o_tup = [dic[a[0]]['dist'] for a in enumerate(atom_array) if a[1] > 8 and a[0] != 0][0] # list of tuples
     if o_tup != []:
         loc = o_tup[np.argmax([a[1] for a in o_tup])][0]
         spin = o_tup[np.argmax([a[1] for a in o_tup])][2]
 elif atom_array[1] == 6:
     loc = dic[1]['dist'][np.argmax([a[1] for a in dic[1]['dist']])][0]
     spin = dic[1]['dist'][np.argmax([a[1] for a in dic[1]['dist']])][2]
 else:
     bad = True

 if not bad:
     if spin == 's1':
         dist = distances.distance_array(traj1[f1][0], traj1[f1][loc], box=box)
         dist2 = distances.distance_array(traj1[f1][1], traj1[f1][loc], box=box)
     else:
         dist = distances.distance_array(traj2[f2][0], traj2[f2][loc], box=box)
         dist2 = distances.distance_array(traj2[f2][1], traj2[f2][loc], box=box)
     dist3 = distances.distance_array(traj2[f2][0], traj2[f2][1], box=box)
     OP = (dist[0][0]-dist2[0][0])/dist3[0][0]
     return dist, OP, spin, loc, bad
 else:
     return None, None, None, None, bad
```