Proton transfer in the photocycle of the photoactive yellow protein
Leenders, E.J.M.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 2

Theory

In this chapter I will introduce the methods I used for my computer simulations and explain their theoretical background. I will start with molecular dynamics (MD) simulations in general: how do we calculate the way particles move? An important aspect of this is the calculation of the forces between particles. Sections 2.2 and 2.3 discuss the two methods used in this thesis: force field and quantum chemical MD respectively. For the force field simulations, I used Gromacs\textsuperscript{35} and Gromos\textsuperscript{36} software, both with the Gromos96 force field.\textsuperscript{31} CPMD\textsuperscript{37,38} and CP2K\textsuperscript{39} software performed the quantum MD. These basic methods have been around for some decades now. More recently - in the last decennary - new techniques have been developed to enable us to get more, or different types of, information from MD simulations. I will explain these ‘advanced methods’ in section 2.4. I will conclude this chapter with a section on how I collect and analyse information from simulation data (section 2.5).

2.1 Molecular dynamics

All molecular dynamics methods in this research use Newton’s laws of motion.

\[
\begin{align*}
F &= ma, \quad (2.1) \\
a &= \frac{dv}{dt}, \quad (2.2) \\
v &= \frac{dr}{dt}, \quad (2.3)
\end{align*}
\]

where the vector \( F \) is the force on a particle, \( a \) its acceleration, \( v \) the velocity and \( r \) the position. \( m \) is the mass of a particle and \( t \) is for time. A different formulation of the equations of motion, widely used in \textit{ab initio} MD, is based on the Lagrangian function. For a system of \( N \) particles of mass \( m_i \) with kinetic energy \( U_{\text{kin}} \) and potential energy \( V \), the Lagrangian \( \mathcal{L} \) is

\[
\mathcal{L}(r^N, \dot{r}^N) = U_{\text{kin}}(\dot{r}^N) - V(r^N) = \sum_{i=1}^{N} \frac{1}{2} m_i \dot{r}_i^2 - V(r^N). \quad (2.4)
\]
One finds the equations of motion by applying the Euler-Lagrange equation

\[ \frac{d}{dt} \frac{\partial L}{\partial \dot{r}_i} = \frac{\partial L}{\partial r_i}, \]  

(2.5)

This leads to

\[ \frac{\partial L}{\partial \dot{r}_i} = m_i \ddot{r}_i, \]  

(2.6)

\[ \frac{d}{dt} \frac{\partial L}{\partial \dot{r}_i} = m_i \dddot{r}_i, \]  

(2.7)

\[ \frac{\partial L}{\partial r_i} = -\frac{\partial V}{\partial r_i}, \]  

(2.8)

hence

\[ \mathbf{F}_i = -\frac{\partial V}{\partial r_i} = m_i \dddot{r}_i, \]  

(2.9)

which for each particle \( i \) corresponds to eq. 2.1. The force on a particle is equal to the derivative of the potential energy. This potential energy is defined by the presence of other particles and/or the presence of an external field. The force changes the velocities \( \mathbf{v} \) of the particles and hence their positions \( \mathbf{r} \). In a new position they feel new forces and the whole cycle starts again. The new position can be found by integrating the equations of motion; in an MD simulation this is done by taking time steps that are so small, that we can assume that the force does not change when the system evolves from \( t \) to \( t + \Delta t \). A simple (though not numerically stable) way is

\[ \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \mathbf{a} \Delta t \]
\[ \mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v} \Delta t. \]  

(2.10)

To ensure energy conservation, proper algorithms are time reversible, just as Newton’s equations are; this algorithm is not. In this work, the software uses two different algorithms\(^\text{34} \) to integrate these equations of motion. CP2K (section 2.3.2) and CPMD (section 2.3.3) use the velocity Verlet algorithm, Gromacs and Gromos (section 2.2) use leapfrog. When starting a simulation, the positions are often known from experiments (for example in case of protein structures) or they can be randomly assigned (for example in case of a gas). The same is true for the velocities, although they are hardly ever experimentally measured for individual particles. An often used approach (also in this thesis) is to draw the velocities from a Maxwell-Boltzmann distribution. The calculation of the forces is explained in subsection 2.1.3.

### 2.1.1 Velocity Verlet algorithm

To explain the velocity Verlet algorithm, I calculate the new position by making a Taylor expansion

\[ \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \mathbf{a} \Delta t \]
\[ \mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v} \Delta t. \]  

(2.10)
\[ \mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \frac{d\mathbf{r}}{dt} \Delta t + \frac{1}{2} \frac{d^2\mathbf{r}}{dt^2} \Delta t^2 + O(\Delta t^3) \]

\[ = \mathbf{r}(t) + \mathbf{v}(t) \Delta t + \frac{1}{2} \mathbf{a}(t) \Delta t^2 = \mathbf{r}(t) + \mathbf{v}(t) \Delta t + \frac{\mathbf{F}(t)}{2m} \Delta t^2. \quad (2.11) \]

For updating the velocities, one assumes that the acceleration during time \( \Delta t \) is constant and based on the average force

\[ \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{F}(t + \Delta t) + \mathbf{F}(t)}{2m} \Delta t. \quad (2.12) \]

Note that in order to calculate the new velocity, the new force and therefore the new position is required.

### 2.1.2 Leapfrog algorithm

The leapfrog algorithm is called this way because position and velocity make ‘leapfrog’ jumps over each other’s back; they are calculated half a time step after each other. If the positions are known at integer time steps, then we define the velocities at half-integer time steps

\[ \mathbf{v}(t - \Delta t/2) \equiv \frac{\mathbf{r}(t) - \mathbf{r}(t - \Delta t)}{\Delta t} \quad (2.13) \]

\[ \mathbf{v}(t + \Delta t/2) \equiv \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t}. \quad (2.14) \]

Eq. 2.14 can be rearranged into

\[ \mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \mathbf{v}(t + \Delta t/2). \quad (2.15) \]

To calculate the new position, the velocity for moving from the old to the new position is calculated from the force at the old position

\[ \mathbf{v}(t + \Delta t/2) = \mathbf{v}(t - \Delta t/2) + \Delta t \frac{\mathbf{F}(t)}{m}. \quad (2.16) \]

A characteristic of the leapfrog algorithm is that the positions and forces are not known at the same moment as the velocities and total energy.

### 2.1.3 Force calculation

From the previous section, it is clear that calculation of the forces between particles is an important issue in all MD simulations. The forces applied to a particle at location \( \mathbf{r} \) depend on the potential energy \( V(\mathbf{r}) \) of the system at that location (according to eq. 2.9). In this thesis I use two main methods to calculate this potential energy and hence the forces: with a force field and \textit{ab initio}. The methods are explained in sections 2.2 and 2.3 respectively.

Calculating forces becomes more demanding when one has more particles in the system, as more terms build up the potential energy. Simulations are most efficient when
they simulate real (bulk) properties with as little particles as possible. To do this, periodic boundary conditions (PBC) are applied to the system. This means that the system is simulated in a box, surrounded by exact copies of this simulation box. Whenever a particle leaves the box on the right side, it enters the neighbouring box, and hence the box itself, on the left side. In this way, boundary effects are avoided and bulk properties can be calculated with a relatively small system. Long-range forces, such as Coulomb forces (see section 2.2), require special attention, as their range usually exceeds the size of the periodic box. Dealing with long-range forces requires correct implementation of the PBC plus efficient handling of potential energies built up out of components from many particles that are all in each other’s interaction range. These forces can be calculated fast and accurately with Ewald summations; this is used in both quantum chemical software packages. Only for non-periodic systems (as in QMMM, see subsection 2.4.1), a Poisson solver is necessary. I use Tuckerman’s algorithm for this. A slightly modified version of the Ewald summation is particle-mesh Ewald, used in the force field MD simulations with Gromacs. The Gromos software uses a different method, the twin-range method.

2.2 Force field molecular dynamics

In force field MD, the potential energy and forces between particles are predefined functions of the position of a particle with respect to the other particles. The functions are empirical and/or based on quantum simulations (see section 2.3). The force field parameters are fitted to reproduce bulk thermodynamics properties, such as the density and solvation enthalpies. We distinguish two types of interactions: bonded and non-bonded potential energies. During the MD simulations, the force is calculated by taking the derivative of the potentials with respect to the position, as in eq. 2.9. A widely used function to describe non-bonded interactions between atom pairs at distance $r_{ij}$ is the empirical Lennard-Jones equation:

$$V_{LJ}(r_{ij}) = 4\epsilon \left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}. \quad (2.17)$$

It combines the repulsive Pauli potential with the attractive Van der Waals interaction. In this equation, $\epsilon$ is the depth of the energy well at the minimum, at a distance of $1.12\sigma$ (see figure 2.1). The parameters depend on the nature of the two particles. Next to this, charged particles feel an electrostatic force. This is calculated from the Coulomb potential for particle $i$ and $j$, depending on their charges $q_i$ and $q_j$ and on the dielectric permittivity $\varepsilon_0$ (which is a different constant from $\epsilon$ in the Lennard-Jones equation):

$$V_{Coul} = \frac{1}{4\pi\varepsilon_0} \frac{q_i q_j}{r_{ij}}. \quad (2.18)$$

The Lennard-Jones potential - and often the Coulomb potential as well - describes forces between particle pairs that are not connected with chemical bonds. When bonds are present, the Lennard-Jones interaction is replaced by so-called bonded interactions. In the rest of this section, I will explain the different types of bonded interactions in my simulations.

Chemical bond stretching between two atoms (or atom groups) on either side of a covalent bond is described with a harmonic potential with spring constant $k_{ij}$ around equi-
2.2 Force field molecular dynamics

Figure 2.1: The Lennard-Jones potential is an empirical description of the repulsive Pauli potential and the attractive Van der Waals interaction. The shape of the potential is defined by $\sigma$ and $\epsilon$.

Equilibrium value $d_{ij}^2$. For computational efficiency, the software uses the squares of the vectors and not the distances (saving many square root operations)

$$V_b(r_{ij}) = \frac{1}{4} k_{ij} (r^2_{ij} - d^2_{ij})^2. \quad (2.19)$$

Three bonded atoms define a bond angle. Its bending vibrations depend on the angle $\theta$ defined by the three atoms $i$, $j$ and $k$

$$V_a(\theta_{ijk}) = \frac{1}{2} k_{ijk} \left( \cos(\theta_{ijk}) - \cos(\theta^0_{ijk}) \right)^2, \quad (2.20)$$

with

$$\cos(\theta_{ijk}) = \frac{r_{ij} \cdot r_{kj}}{r_{ij} r_{kj}}. \quad (2.21)$$

Dihedral potentials describe the interaction between four neighbouring atoms. There are two types. Improper dihedrals (eq. 2.22) keep planar groups - such as aromatic rings - planar or make sure that chiral centres do not change into their mirror image. They are defined by the angle $\xi$ between the two planes defined by the atoms $ijk$ and $jkl$, according to figure 2.2 and eq. 2.22

$$V_{id}(\xi_{ijkl}) = \frac{1}{2} k_\xi (\xi_{ijkl} - \xi_0)^2. \quad (2.22)$$

Proper dihedrals regulate rotation around bonds, as in figure 2.2d,

$$V_d(\xi_{ijkl}) = k_\xi (1 + \cos(n\xi_{ijkl} - \xi_0)). \quad (2.23)$$

In the case of a double bond, $n = 2$, as there are two possible low-energy configurations. In the cis configuration, atom $i$ and $l$ are on the same side and $\xi = 0$; in the trans configuration the angle $\xi = 180^\circ$. By choosing $\xi_0$ equal to one of these angles, $V_d(\xi_{ijkl})$ shows two equal minima when the atoms are in one of these configurations.

In this thesis I use the Gromos96 force field\textsuperscript{31} - version 43A1 - to describe these interactions. A newer version became available during my PhD, but I have not used it, in order to be able to compare my work to my own older work. I also use work based on version
Theory

Figure 2.2: Definition of dihedrals in the force field according to equations 2.22 and 2.23. ‘a’ to ‘c’ are improper dihedrals; ‘a’ shows a part of an aromatic ring, ‘b’ an atom that is surrounded by three others in a flat configuration and ‘c’ is a chiral centre. ‘d’ defines a proper dihedral around a double bond.

43A1 from colleagues as a starting point for my own simulations (see chapter 6). The force field provides the parameters to be used in eqs. 2.17 to 2.23 for calculating the non-bonded interaction, bond stretching, bending vibrations and dihedral interactions respectively. It is fitted to describe proteins and their solvents. The force field also includes (partial) charges $q_i$ on atoms, so the Coulomb force can be calculated. The Gromos96 force field works with implicit hydrogens, meaning that non-polar hydrogens are not included in the simulation. Instead, for example a CH$_2$ group is treated as one particle. The Gromacs developers advise to use the Gromos96 force field together with the simple point charge (SPC) description of water molecules where all three atoms in a molecule are restrained by harmonic bond potentials and have a partial charge of $0.82\, e$ for the oxygen atom and $0.41\, e$ for both hydrogens.

2.3 Ab initio molecular dynamics

Not all situations and events are described well with force field MD. Examples are excited states, some dipole-dipole interactions and chemical reactions. In all these cases, the forces between atoms depend on their positions, on time, on polarising groups nearby, on external fields etc. It no longer suffices to regard atoms as simple particles with a fixed charge, so we have to take into account that they consist of nuclei surrounded by electrons. In this way, the forces are calculated ab initio, without any predefined force field parameters, on the fly. More precise, quantum chemical theory calculates how the electron density is influenced by the positive nuclei in the system and vice versa, how the spreading of negative charge moves the positive particles. Theoretically, apart from the starting positions of the atomic cores and the laws of quantum mechanics, nothing else is needed to perform the simulation. A system of nuclei and electrons is described by the Schrödinger equation. Solving this equation is computationally very demanding, as the number of variables is much larger than in force field simulations. Not only do we have degrees of freedom for every atom, every electron now has degrees of freedom that have to be calculated too.

The rest of this section is mainly based on the very complete and well-written textbook by Martin. I will ignore the electron spin, as in my work I only have orbitals that contain two indistinguishable electrons.

The basic formulation of the time-independent Schrödinger equation is

$$\hat{H}\psi(\mathbf{r}_i, \mathbf{R}_j) = E\psi(\mathbf{r}_i, \mathbf{R}_j),$$  \hspace{1cm} (2.24)
with $\psi(r_i, R_j)$ the wavefunction describing the complete system based on the spatial coordinates of electrons ($r_i$) and ions ($R_j$). Applying the Hamilton operator $\hat{H}$ on $\psi$ gives us the eigenvalue $E$, the total energy of the system. For a system of electrons (lower case) and ions (upper case), the full Hamiltonian is

$$\hat{H} = -\frac{\hbar^2}{2m_e} \sum_i \nabla_i^2 - \sum_{i,I} \frac{Z_I e^2}{4\pi\epsilon_0 |r_i - R_I|} + \frac{1}{2} \sum_{i \neq j} \frac{e^2}{4\pi\epsilon_0 |r_i - r_j|}$$

$$- \sum_I \frac{\hbar^2}{2M_I} \nabla_I^2 + \frac{1}{2} \sum_{I \neq J} \frac{Z_I Z_J e^2}{4\pi\epsilon_0 |R_I - R_J|},$$

with $\hbar = 2\pi\hbar$ Planck’s constant, $m_e$ and $M_I$ the masses of an electron and ion $I$ respectively and $Z_I$ the atomic number.

The first term is the kinetic energy of the electrons, the second term the electron-ion Coulomb interaction, then the electron-electron Coulomb energy, the ionic kinetic energy and finally the potential energy due to ion-ion Coulomb interaction.

Up to the system of two protons with one electron (positively charged hydrogen molecule), the eigenvalue problem of eq. 2.24 can be solved analytically. For larger systems, we need to use approximations and numerical methods. Different approaches have been proposed to tackle this question; I refer to any text book on electronic structure calculations for an overview.

A very good approximation in almost every ground state system is the Born-Oppenheimer or adiabatic approximation. For calculating the energy of electrons in the ground state, it is reasonable to assume that the positions of the nuclei are fixed and hence the kinetic energy of the nuclei can be eliminated from the Hamiltonian. The idea behind this is that ions are so heavy with respect to electrons that you can decouple their motion. This leaves us with the Hamiltonian for the electronic structure

$$\hat{H} = \hat{T} + \hat{V}_{ext} + \hat{V}_{int} + E_{II}.$$  

(2.26)

$\hat{T}$ is the operator giving the kinetic energy of the electrons, $\hat{V}_{ext}$ gives their potential energy due to the external field of the ions, $\hat{V}_{int}$ the electron-electron interaction and the constant term $E_{II}$ is the classical Coulomb interaction of the nuclei. As this last term does not change, it can be left out.

In dynamical quantum simulations, the method of choice nowadays for calculating the electronic structure is density functional theory. It is explained in section 2.3.1. Sections 2.3.2 and 2.3.3 present two algorithms that are the basis for the $ab$ initio software used for this thesis.

### 2.3.1 Density functional theory

Density functional theory (DFT) calculates the energy of a (molecular) system using its electron density. DFT became practically useful in 1964, when Hohenberg and Kohn proved that the ground state energy of a system has a one-to-one relation with the electron density distribution. They were the first to formulate DFT exactly, using two theorems. I) In a system of interacting electrons in an external potential $V_{ext}$, this $V_{ext}$ is determined uniquely,
except for a constant, by the ground state electron density $\rho_0$; II) For any $V_{\text{ext}}$, there is a universal energy functional $E[\rho]$, depending on the electron density. By minimising this functional (applying the variational principle), the ground state energy and ground state density are found. This Hohenberg-Kohn energy functional has the following form

$$E_{HK}[\rho] = T[\rho] + E_{\text{int}}[\rho] + \int dV_{\text{ext}}(r)\rho(r)$$

$$\equiv F_{HK}[\rho] + \int dV_{\text{ext}}(r)\rho(r).$$  \hspace{1cm} (2.27)

The electron density is a function of $r$, with $\rho(r) \geq 0$. Integrated over the entire space, it must be equal to the total number of electrons. $F_{HK}[\rho]$ is ‘universal’ in the sense that it does not depend on the position of the nuclei. The interaction of the electrons with the ‘external’ potential of the nuclei is represented by the integral term in eq. 2.27.

From the second theorem, the ground state energy $E_0$ is

$$E_0 = E[\rho_0(r)] \leq E[\rho(r)],$$  \hspace{1cm} (2.28)

with $\rho_0$ the ground state density. Applying the variational principle will give us the minimum energy, under the constraint that the total number of electrons is constant. This result is given by the Euler-Lagrange equation, with Lagrange multiplier $\mu$ being the chemical potential of the electrons

$$\mu = \frac{\partial E[\rho(r)]}{\partial \rho(r)} = V_{\text{ext}}(r) + \frac{\partial F_{HK}[\rho(r)]}{\partial \rho(r)}.$$  \hspace{1cm} (2.29)

The Hohenberg-Kohn equation, eq. 2.27, is exact. But as long as the precise energy functional is not known, it is not particularly useful. The Kohn-Sham ansatz provides a way of describing the functional. It introduces a reference system, for which the functional is easier to solve than for the interacting many-body system. A crucial factor is the exchange-correlation functional, that incorporates the difference between the real and the reference system. The ansatz now is that the ground state density of the real system can be represented by the ground state density of an auxiliary system $S$ of non-interacting electrons.48

The density of the auxiliary system is built up by the squares of the so-called Kohn-Sham orbitals $\phi_i$. Although these orbitals are not the ‘real’ orbitals, they are often used as a good representation and add up to the real density

$$\rho(r) = \sum_{i=1}^{N} |\phi_i(r)|^2,$$  \hspace{1cm} (2.30)

where $2N$ is the number of electrons in the system ($2N$ electrons fill $N$ orbitals). In a simulation, the orbitals will be built up from a suitable basis set. The kinetic energy of these non-interacting electrons is

$$T_S = -\frac{\hbar^2}{2m_e} \sum_{i=1}^{N} <\phi_i|\nabla_i^2|\phi_i>$$  \hspace{1cm} (2.31)

The Coulomb interaction between the electrons is calculated as the Hartree energy.
\[
E_{\text{Hartree}}[\rho] = \frac{e^2}{2} \int \text{d}^3r \text{d}^3r' \frac{\rho(r)\rho(r')}{4\pi\epsilon_0 |r - r'|}.
\] (2.32)

Note that this is not exactly the same as the \( V_{\text{int}} \) term in eq. 2.26, as eq. 2.32 includes an extra self-interaction term. Because the integration runs over the complete density twice, all electrons necessarily interact with themselves. I will come back to this when I discuss the exchange-correlation energy \( E_{\text{XC}} \). The potential field from the nuclei now interacts with the electron density instead of the separate electrons. For any point in an external field of \( N \) ions, \( V_{\text{ext}} \) is

\[
V_{\text{ext}}(r) = -\sum_{I=1}^{N} \frac{Z_I e}{4\pi\epsilon_0 |r - R_I|}.
\] (2.33)

The energy functional is now

\[
E_{\text{KS}} = T_S + \int \text{d}^3r V_{\text{ext}}(r)\rho(r) + E_{\text{Hartree}}[\rho] + E_{\text{XC}}[\rho].
\] (2.34)

The exchange-correlation energy \( E_{\text{XC}} \) is the difference between all other terms and the Hohenberg-Kohn equation (eq. 2.27), hence

\[
E_{\text{XC}}[\rho] = F_{\text{HK}}[\rho] - (T_S + E_{\text{Hartree}}[\rho]).
\] (2.35)

\( E_{\text{XC}} \) includes the difference in the kinetic energy of the real and the auxiliary system and the difference between the Hartree energy and the real electron-electron interaction. If \( E_{\text{XC}} \) is known exactly, then \( E_{\text{KS}} \) is exact. However, in practice it must be estimated. When this estimation is sufficiently good, then the Kohn-Sham approach provides reasonable results.

Easy and widely-used is the Local Density Approximation (LDA),

\[
E_{\text{XC}}^{\text{LDA}} = \int \text{d}r \rho(r) \varepsilon_{\text{xc}}([\rho(r)], r),
\] (2.36)

with \( \varepsilon_{\text{xc}} \) the exchange-correlation energy per electron in a homogeneous electron gas at constant density. This \( \varepsilon_{\text{xc}} \) is calculated and tabulated for many densities with very accurate quantum Monte Carlo simulations.

The simple LDA works surprisingly well for solid state calculations. However, a quantitative study of chemical reactions is problematic, as a high accuracy of a few kcal/mol is needed. In this case, a generalised gradient approximation (GGA) improves the LDA. It adds extra terms to (2.36) depending on derivatives of the density. A well-known example of this is the BLYP functional; it is used throughout this thesis. It combines the exchange energy approximation of Becke\(^49\) with the correlation energy approximation of Lee, Yang and Parr.\(^50\)

### 2.3.2 Born-Oppenheimer molecular dynamics

In the next two subsections, I discuss the way in which DFT is implemented in dynamical ab-initio software,\(^51\) starting with the Born-Oppenheimer (BO) algorithm. The potential

---

\(^*\)I made use of some very useful lecture notes by Jürg Hutter. Notes of many of his classes are available on internet.
energy is the Kohn-Sham energy, leading to the Lagrangian

$$\mathcal{L}_\text{BO}(\mathbf{R}^N, \dot{\mathbf{R}}^N) = \sum_{i=1}^{N} \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 - \min_{\{\phi_i\}} E^{KS}(\{\phi_i\}; \mathbf{R}^N). \tag{2.37}$$

Under the condition that the minimisation of $E^{KS}$ leads to orthogonal wavefunctions $\phi_i$, the Euler-Lagrange equation gives us

$$M_I \ddot{\mathbf{R}}_I = -\nabla_I \left( \min_{\{\phi_i\}} E^{KS}(\{\phi_i\}; \mathbf{R}^N) \right). \tag{2.38}$$

This is equal to the Hellmann-Feynman theory,\textsuperscript{52} which gives us the force on ion $I$ for a system with a known wavefunction $\psi$

$$F_I = -\frac{\partial E}{\partial \mathbf{R}_I} = -\langle \psi | \frac{\partial \hat{H}}{\partial \mathbf{R}_I} | \psi \rangle. \tag{2.39}$$

As we use orthogonality constraints ($\Lambda_{ij}$, see also subsection 2.4.2) for the orbitals, this leads to an extended energy functional to be minimised

$$\mathcal{E}^{KS} = E^{KS} + \sum_{ij} \Lambda_{ij} (\langle \phi_i | \phi_j \rangle - \delta_{ij}). \tag{2.40}$$

After minimisation, the forces on the ions are then calculated by differentiating eq. 2.40 with respect to $\mathbf{R}_I$, according to eq. 2.38. From the force we calculate the new ionic velocities, new ion positions, new ground state energy, new force, etc.

The BOMD software in this work is Quickstep,\textsuperscript{53} part of the CP2K package.\textsuperscript{39} It is very efficient in optimising the energy functional, because it uses a basis set of plane waves and Gaussians. Its CPU usage scales as $O(MN^2)$,\textsuperscript{54} with $M$ the number of basis set functions and $N$ the number of electrons.

### 2.3.3 Car-Parrinello molecular dynamics

Car and Parrinello are the pioneers in the field of quantum molecular dynamics. Their 1985 paper\textsuperscript{55} gave this field of research an enormous boost; it gained even more importance with increasing computer power. The basic idea is that they do not treat the electrons and ions separately, as in BOMD, but as one unified problem.\textsuperscript{45} Hence, the time evolution of the electronic ground state is also solved within the molecular dynamics algorithm. This requires the introduction of a fictitious kinetic energy and fictitious mass $\mu$ to the electrons, resulting in the following combined Lagrangian for the ions and electrons

$$\mathcal{L} = \sum_{i=1}^{n} \frac{1}{2} (2\mu) \langle \dot{\phi}_i(\mathbf{r}) | \dot{\phi}_i(\mathbf{r}) \rangle + \sum_{I=1}^{N} \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 - V(\{\phi_i\}, \mathbf{R}^N)$$

$$+ \sum_{ij} \Lambda_{ij} (\langle \phi_i(\mathbf{r}) | \phi_j(\mathbf{r}) \rangle - \delta_{ij}). \tag{2.41}$$

$n$ is the number of orbitals, containing $2n$ electrons. The last term ensures orthogonality of the electronic states. Applying the Euler-Lagrange equation with respect to $\mathbf{R}_I$ gives
exactly the same equation of motion for the ions as in BOMD, eq. 2.38. However, for CPMD we have to do this for the orbitals as well

\[ \mu \ddot{\phi}_i (r, t) = -\frac{\partial E}{\partial \langle \phi_i (r) \rangle} + \sum_j \Lambda_{ij} \langle \phi_j (r, t) \rangle. \] (2.42)

The fictitious mass \( \mu \) needs to be chosen in such a way that the lowest electronic frequency is much larger than the highest frequency of the nuclei, to prevent energy transfer. The time step has to be compatible with the highest electronic frequency.\(^{56}\) In BOMD, the highest ionic frequency governs the time step.

In CPMD, wavefunctions are expanded in a plane wave basis set. Each wavefunction is a linear combination of orthonormal plane waves \( \chi \). Instead of the time evolution of the wavefunction in time, the evolution of the parameters \( c_k \) is calculated

\[ \phi_i (r) = \sum_k c^i_k \chi_k (r) \] (2.43)

\[ \mu \frac{\partial^2 c^i_k}{\partial t^2} = -\frac{\partial E}{\partial c^*_{ik}} + \sum_j \Lambda_{ij} c^j_k. \] (2.44)

The expansion is truncated after a given number of plane waves, corresponding to the cut-off energy \( E_{\text{cut}} \). In order to describe the wavefunction close to the nuclei correctly, this \( E_{\text{cut}} \) should be extremely high (including many plane waves); however, this is not necessary for the rest of the simulation box and the core electrons are not even interesting from a chemical point of view. Hence, this basis set is almost always combined with the use of pseudopotentials. Applying the frozen core approximation, the effect of the nucleus and the core electrons on the valence electrons is described with a pseudopotential.\(^{57}\) This potential is much smoother than the actual potential inside the core region, but the two potentials are the same in the valence region. This is particularly useful in combination with a plane wave basis set, as is used in CPMD.

A pseudopotential can be generated from an all-electron atomic simulation. It has to fulfil four conditions in order to be “norm-conserving.”\(^{58}\)

- The valence pseudo wavefunctions generated using the pseudopotential should contain no nodes. Smoother wavefunctions are easier to describe with a plane wave basis set.
- Beyond the cut-off radius \( r_{cl} \) the normalised atomic radial pseudo wavefunction (PP) with angular momentum \( l \) is equal to the all-electron (AE) wavefunction

\[ R_{l}^{PP} (r) = R_{l}^{AE} (r), \quad r > r_{cl}, \] (2.45)

or converges rapidly.
- The charge enclosed within \( r_{cl} \) must be equal for the two wavefunctions.
- The eigenvalues must be equal, \( \varepsilon_{l}^{PP} = \varepsilon_{l}^{AE} \).

With the pseudo wavefunction known, the pseudopotential is found by using the inverse Schrödinger equation.
The RATTLE algorithm\textsuperscript{59} calculates the Lagrange multipliers $\Lambda_{ij}$ of eq. 2.44 within the velocity Verlet integrator. (SHAKE\textsuperscript{60} does the same thing in normal Verlet.) These constraints are handled in the same way as any other geometrical constraint within CPMD. This is discussed in subsection 2.4.2.

2.4 Advanced methods

The main drawback of quantum molecular dynamics is the computational power it demands. In CPMD the time step is very small, in BOMD an energy minimisation needs to be performed every time step. In both methods, applying the Hamiltonian is quite demanding. This constrains quantum MD simulations to system sizes of several hundreds of ions and to simulation times of the order of ten picoseconds nowadays. To be able to find answers to chemically relevant questions, we have to turn to smart methods that help us minimise the computational costs and maximise the information we get from our calculations.

2.4.1 Quantum mechanics/molecular mechanics

Quantum mechanics/molecular mechanics (QMMM) helps us to simulate much larger system sizes; however, it does not change the simulation times that can be reached. The method does this by using quantum methods only for the region where it is necessary. The rest of the system is treated with force field MD. Its main use is in the field of biochemistry; this thesis is an example of that. Many chemical reaction in proteins or DNA happen locally, but the (changing) structure of the biomolecule does influence the reaction environment. Only where the electron density changes drastically (for example when bonds break), a quantum description is useful. For the rest of the molecule (and its solvent and possible other substances present) a force field description suffices. However, it is important to take into account the interactions between the atoms in the two different descriptions properly: How do you describe a bond between a quantum and a force field atom? How does a point charge interact with an electron density in a computationally fast way? The longer the distance between the quantum part and a force field atom, the more insignificant the exact electron density is. The QMMM software used in this thesis\textsuperscript{61} provides an interface layer between CPMD and Gromos software (see figure 2.3).

The two main issues that this interface has to solve are the bonded and non-bonded interactions. The former only have to be taken into account when the QM/MM boundary cuts through a chemical bond. In this thesis, this is in between two non-polar carbon atoms. To ensure the correct valence orbital structure for the QM part, the first MM carbon is replaced by a capping hydrogen, which is only seen by the QM part. The MM carbon ‘sees’ the normal force field bond.

The non-bonded part of the Hamiltonian for MM particles $I$ interacting with QM particles $j$ is\textsuperscript{32}

\begin{equation}
H_{\text{non-bonded}} = \sum_{I \in \text{MM}} q_I \int \frac{\rho(r)}{4\pi\varepsilon_0 |r - r_I|} + \sum_{I \in \text{MM}, j \in \text{QM}} v_{\text{vdw}}(r_{ij}).
\end{equation} (2.46)
When QM particles interact with MM particles, they have a Van der Waals radius from the force field. The charge of the QM part is taken into account by D-RESP charges. These dynamically-generated restrained electrostatic potential charges are fitted every time step with a quadratic penalty function to their corresponding Hirschfeld charges. In this way, they show less fluctuations than unconstrained ESP charges, and hence show less electron spill-out: the problem of electronic density from the QM part moving towards positively charged MM particles, that lack Pauli repulsion. With a least-squares fit, the D-RESP charges reproduce the electric field that nearby MM atoms in a shell of thickness \( r_c \) around the QM part feel. The long-range interactions between the D-RESP charges and the partial MM charges are calculated with a quadrupolar expansion.

### 2.4.2 Constrained dynamics

In constrained dynamics, one or more degrees of freedom in a molecular dynamics simulation are constrained to a fixed value. This can be the position of an atom, a distance between two atoms, or the difference between two distances, etc. Constraints are common in force field calculations, where the bond lengths and angles are fixed (or at least restrained). In this thesis, I also apply constraints to some bonds in QM simulations. Realising that this influences the dynamics in an unphysical way, constrained dynamics simulations can still teach us something about systems that we cannot study without these constraints. In large systems, such as the systems studied in this thesis, constraints can help to focus only on a specific part of phase space. From differences in the constraint force, the force that is needed to keep the constraint, one can also calculate free energy differences.

Any geometrical constraint \( \sigma(R^N_I) \) can be obeyed within the velocity Verlet integrator with RATTLE:

\[
\sigma(\{R^N_I\}) = 0. \tag{2.47}
\]

For example, in case of a distance constrained at \( d_{ij} \), eq. 2.47 would be
\[ |R_I - R_J| - d_{IJ} = 0. \]  
(2.48)

The \( i \) constraints add an extra term to the Lagrangian

\[ \mathcal{L} = T(v_N(t)) - V(R_N(t)) + \sum_i \lambda_i(t)\sigma_i(R_N(t)). \]  
(2.49)

This will add an extra term to the force as well, the constraint force \( g(t) \). For any particle in the system this is

\[
\begin{align*}
ma &= -\frac{\partial V}{\partial R} + \sum_i \lambda_i \frac{\partial \sigma_i}{\partial R}, \\
g(t) &= -\sum_i \lambda_i \nabla \sigma_i.
\end{align*}
\]  
(2.50) (2.51)

The Lagrange multipliers \( \lambda_i \) (one for each constraint) have to be determined in such a way that the value of the constraint does not change with the next time step. This gives a slightly modified version of eq. 2.11

\[ r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{F(t) + g(t)}{2m} \Delta t^2. \]  
(2.52)

With an iterative process, we could fit \( g(t) \) to make sure that the constraints are met. However, we run into a problem for the modified equation for the velocity (based on eq. 2.12)

\[ v(t + \Delta t) = v(t) + \frac{F(t + \Delta t) + g(t + \Delta t) + F(t) + g(t)}{2m} \Delta t. \]  
(2.53)

Solving this is not possible, because we need to know \( g(t + \Delta t) \) in order to calculate \( v(t + \Delta t) \); in other words, we need \( g(t) \) to calculate \( v(t) \), but we only get to know \( g(t) \) one time step later, when we solve eq. 2.11 to find \( r(t + \Delta t) \). The idea behind RATTLE is that this problem can be solved by using two different \( g \)'s and hence two sets of Lagrange multipliers: \( g_r(t) \) to fit eq. 2.52 and an extra \( g_v(t) \) to fit eq. 2.53.59

### 2.4.3 Metadynamics

With metadynamics, one can explore a multidimensional free energy surface (FES) efficiently. First, one has to define one or more collective variables that define the region of phase space for which the FES is needed. In the first paper on this method, Laio and Parrinello65 defined an extra coarse-grained dynamics of these variables, hence the name of the method. The basic idea is simple: when doing a normal MD simulation in a system that has a FES with several minima and high barriers in between, the system will stay in one minimum (at least within a realistic simulation time). With metadynamics, one adds small free energy hills to this landscape on the locations (defined by the collective variables \( s_i \)) that have already been sampled. In this way, the regions that are sampled will get a higher free energy. Aiming to minimise its free energy, the system will evolve to regions that have not been sampled yet. When the simulation is sufficiently long, all regions in the phase space will be visited. In addition, it can be shown that the sum of all the small
hills that have been added is equal to the inverse shape of the FES. When a metadynamics simulation is equilibrated, the total FES (that is, the original FES plus the bias potential) is completely flat, and all locations in the order parameter space are equally likely to be visited. Note however that the MD simulation is now non-Markovian: the FES, and hence the force on the particles, is history dependent.

The extra free energy terms are usually Gaussians with some height $H$ and width $\delta \sigma$. The equations are written in the scaled collective variable $\sigma_i = s_i / \Delta s_i$, with $\Delta s_i$ a constant of roughly the width of the valley in that variable. The scaled force at time $t$, $\phi^i_t = F^i_t / \Delta s^i_t$, then includes an extra term in addition to the normal MD force components: the negative derivative of the sum of the Gaussians

$$ \phi_i^{(meta)} = \phi_i^{(MD)} - \frac{\partial}{\partial \sigma_i} H \sum_{t' \leq t} \prod_i \exp \left( -\frac{|\sigma_i - \sigma_i^{t'}|^2}{2 \delta \sigma^2} \right). $$

(2.54)

After a sufficiently long simulation of time $t$ and when the Gaussians are sufficiently small with respect to free energy differences in the FES, the bias potential (multiplied by -1) converges to the ‘real’ free energy $F$

$$ F(s) = - \sum_{t' \leq t} H \exp \left( -\frac{|\sigma - \sigma^{t'}|^2}{2 \delta \sigma^2} \right). $$

(2.55)

The height, width and scaling factor of the Gaussians depend on the system one studies and have to be chosen appropriately. There are two ways in which metadynamics is implemented in the CPMD software: direct and Lagrangian metadynamics. When one uses direct metadynamics, one defines a certain timespan after which a new Gaussian is deposited at the current location. With Lagrangian metadynamics, the Gaussians are deposited at the location of fictitious particles $s_\alpha$ (one for each collective variable) that have a mass $\mu_\alpha$, a force constant $k_\alpha$ restraining the particle to its collective variable $S_\alpha(r)$, and their own metadynamics, giving the augmented Lagrangian

$$ L = L_{MD} + \sum_\alpha \frac{1}{2} \mu_\alpha \dot{s}_\alpha - \sum_\alpha \frac{1}{2} k_\alpha (S_\alpha(r) - s_\alpha)^2 - V(t, s), $$

(2.56)

with $V(t, s)$ the current FES based on the added Gaussians. In this thesis, I use only direct metadynamics.

2.5 Analysis

In this thesis, I use several methods to analyse my computational results. Most of them are intuitive, but some need further explanation. I discuss three methods here that I use.

2.5.1 Radial distribution function

The radial distribution function $g(r)$ gives the relative density of particles around one or more other particles as a function of the distance between these particles. It is calculated by counting the number of particles $N$ in a spherical shell with radius $r$ and thickness $dr$.
around a particle and dividing this by the average density \( \rho \) in the system multiplied by the volume of the shell

\[
g(r) = \frac{\rho_{\text{local}}}{\rho} = \frac{N}{4\pi r^2 \, \text{d}r \rho}.
\] (2.57)

The value of \( r \) cannot exceed half the (minimal) simulation box length, to avoid counting particles twice.

### 2.5.2 Mulliken charges

Ab initio methods calculate the electron density around certain ions. This density can be broken down into partial charges, assigned to each atom in the molecule. Mulliken described one way to do this. To explain his method, I will first show how it works for a molecule consisting of two atoms that both contribute one atomic orbital (AO) to form a molecular orbital (MO); after that I will expand it to any number of atoms and orbitals. This explanation is based on Mulliken’s paper.\(^6^7\)

From the simulation, one knows which AOs (which can be hybrids combining several basis set functions) build up an MO. In this example, we take the normalised AOs \( \chi_r \) and \( \chi_s \) from atoms \( k \) and \( l \) that combine linearly to form MO \( \phi \)

\[
\phi = c_r \chi_r + c_s \chi_s.
\] (2.58)

The spatial distribution of the electron density is obtained from the square of \( \phi \).

\[
n\phi^2 = n c_r^2 \chi_r^2 + 2 n c_r c_s S_{rs} (\chi_r \chi_s / S_{rs}) + n c_s^2 \chi_s^2.
\] (2.59)

Here, \( n \) is the number of electrons in the MO (usually 2) and \( S_{rs} \) the overlap integral. The AO’s are normalised, so upon integration over all space, many terms become unity

\[
n = n c_r^2 + 2 n c_r c_s S_{rs} + n c_s^2.
\] (2.60)

As the overlap term is symmetrical with respect to the two atomic centres, half of it is assigned to atom \( k \) and the other half to atom \( l \). If we then assign the electron density of this MO to the two atoms, we get

\[
n(k) = n(c_r^2 + c_r c_s S_{rs}); \quad n(l) = n(c_s^2 + c_r c_s S_{rs}).
\] (2.61)

By repeating this process for every MO \( i \) in which an AO \( r \) of atom \( k \) takes part, the (partial) number of electrons on this atom is found

\[
n(k) = \sum_i n(i) c_{ir_k} \left( c_{ir_k} + \sum_{l \neq k} c_{is_l} S_{rs_l} \right).
\] (2.62)

The partial charge \( q(r_k) \) on atom \( k \) is then the difference between \( n(r_k) \) and the total number of electrons \( n_0 \) on \( k \) when it is a free neutral atom in its ground state

\[
q(k) = n_0(k) - n(k).
\] (2.63)
2.5 Analysis

2.5.3 Free energy / statistical mechanics

Statistical mechanics teaches us that there is a relation between the Helmholtz free energy $F$ of a system with a fixed number of particles at temperature $T$ and the probability $P_i$ of finding this system at a quantum state $i$.\(^{34}\) This can be understood from the thermodynamical definition of $F$

$$F = U - TS$$

$$\frac{F}{T} = \frac{U}{T} - S$$

$$\frac{\partial}{\partial \frac{1}{k_B T}} \left( \frac{F}{T} \right) = U = \langle E \rangle. \quad (2.64)$$

The internal energy $U$, or in a statistical approach the average microscopic internal energy $\langle E \rangle$, is known from the energy distribution (here formulated in terms of quantum states)

$$\langle E \rangle = \sum_i E_i P_i = \frac{\sum_i E_i \exp(-E_i/k_B T)}{\sum_i \exp(-E_i/k_B T)} = \frac{\partial \ln \sum_i \exp(-E_i/k_B T)}{\partial 1/k_B T}$$

$$= -\frac{\partial \ln Q}{\partial 1/k_B T}. \quad (2.65)$$

The last line defines the partition function $Q$. Combining eqs. 2.64 and 2.65 gives

$$F = -k_B T \ln Q = -k_B T \ln \left( \sum_i \exp(-\beta E_i) \right). \quad (2.66)$$

Remember that $\beta = 1/k_B T$. In practice, free energy differences are more useful to work with than absolute values. An example is the calculation of the free energy change when moving along a reaction coordinate $\lambda$. In that case, bins are formed for small regions in $\lambda$ and $F(\lambda)$ is calculated per bin. Any constant $C$ can be added to eq. 2.66 when doing this, as only the difference between the bins is of interest.

$$F(\lambda) = -k_B T \ln \left( \sum_i \exp(-\beta E_i(\lambda)) \right) + k_B T \ln C$$

$$= -k_B T \ln \left( \frac{\sum_i \exp(-\beta E_i(\lambda))}{C} \right)$$

$$F(\lambda) - F(\lambda_0) = -k_B T \ln \left( \frac{\sum_i \exp(-\beta E_i(\lambda))}{\sum_i \exp(-\beta E_i(\lambda_0))} \right) = -k_B T \ln \left( \frac{P(\lambda)}{P(\lambda_0)} \right). \quad (2.67)$$

The free energy difference is found by counting the number of states in every bin and dividing it by the number of states in one specific bin $\lambda_0$, for example the one with the highest occupancy $P(\lambda_0)$ and hence the lowest free energy. The free energy of this specific bin will then be set to $F = -k_B T \ln 1 = 0$ by choosing $C$ appropriately. For all other values of $\lambda$ the relative free energy can be calculated.