Implementation and Application of QM/MM Hybrid Methods

Dissertation zur Erlangung
des naturwissenschaftlichen Doktorgrades
der Julius-Maximilians-Universität Würzburg

vorgelegt von

Susanne Sauer

geboren in Schweinfurt

Würzburg 2020

Eingereicht bei der Fakultät für Chemie und Pharmazie am

____

Gutachter der schriftlichen Arbeit

1. Gutachter: ____________________________

2. Gutachter: ____________________________

Prüfer des öffentlichen Promotionskolloquiums

1. Prüfer: ____________________________

2. Prüfer: ____________________________

3. Prüfer: ____________________________

Datum des öffentlichen Promotionskolloquiums

____

Doktorurkunde ausgehändigt am

____

Inhaltsverzeichnis

1 Introduction 1

2 Aim of the Work 3

3 Theory 5

3.1 QM/MM 6

3.1.1 Additive QM/MM 7

3.1.2 Subtractive QM/MM 7

3.1.3 Treatment of bonds between QM and MM region 8

3.1.4 Treatment of the Electrostatic Interactions 9

3.1.5 Comparison of Additive and Subtractive Scheme 13

3.2 Density-Functional Tight-Binding (DFTB) 14

3.2.1 Density Functional Theory (DFT) 14

3.2.2 The Tight-Binding Approximation 16

3.2.3 Energy Contributions 18

3.2.4 Calculation of the DFTB-Energy 22

3.2.5 Application 26

3.3 Free Energy Calculation 27

3.3.1 Potential of Mean Force (PMF) 29

3.3.2 Umbrella Sampling 29

3.3.3 Weighted Histogram Analysis Method 31

3.3.4 Example: Rotation of Butane 32

3.3.5 Potential of Mean Force with Interpolation Correction 35

4 Implementation of QM/MM 37

4.1 Additive QM/MM 37

4.1.1 Treatment of Bonded Interactions 39

4.1.2 Treatment of the Electrostatic Interactions 42

4.1.3 Summary: Algorithm 46

4.2 Subtractive QM/MM 48

4.2.1 Three-Layer Scheme 52

4.2.2 Multicenter QM/MM 56

4.3 Periodic Boundary Conditions in QM/MM 57

4.3.1 General Introduction to Periodic Boundary Conditions 58

4.3.2 Implementation of PBC into QM/MM 61

5 Further Implementation 67

5.1 External Energy Interfaces 67

5.1.1 Implementation Details 68

5.1.2 User-defined Options for DFTB$^+$ 69

5.1.3 User-defined Options for Orca 71

5.2 Structure Input and Output Formats 72

5.2.1 The XYZ Format 72

5.2.2 The PDB Format 77
5.3 Extensions of the Umbrella Sampling Task 81
5.3.1 Two-dimensional Umbrella Sampling 82
5.3.2 Linear Combinations of Distances as Reaction Coordinate 83
5.3.3 Potentials of Mean Force with Interpolated Corrections 87
5.4 Optimization Methods 94
5.4.1 Microiterations in QM/MM 94
5.4.2 OPT++ Optimizer for Constraint Optimization 100

6 Investigation of the Reaction Path of Rhodesain and K11777 107

6.1 Benchmark of the QM/MM Methods 110
6.1.1 Comparison to Gaussian-ONIOM 110
6.1.2 Influence of Different Options 111
6.1.3 Influence of the QM Region 116
6.2 2D Scans 119
6.2.1 Local Optimizations 119
6.2.2 2D Scans from Covalent Complex 121
6.2.3 2D Scans from Non-Covalent Complex 122
6.3 Umbrella Sampling 126
6.4 Molecular Dynamics Simulation 131

7 Summary 137

8 Zusammenfassung 139

9 Danksagung 141

Literatur 143

A Appendix 155

1 Introduction

“The Alliance of Newton’s Apple and Schrödinger’s Cat” – This is how the Nobel Prize in Chemistry 2013 has been described.[1] It was awarded to Martin Karplus, Michael Levitt, and Arieh Warshel for the development of QM/MM, which allows for a meaningful combination of quantum mechanics (Schrödinger’s Cat) and classical mechanics (Newton’s Apple).[1]

The first step towards QM/MM was taken in 1972, when Warshel and Karplus published the calculation of 1,6-diphenyl-1,3,5-hexatriene, a planar molecule.[2] For this purpose, they joined their previous experience in quantum mechanics (Karplus) and molecular mechanics (Warshel).[1] The π electrons of the molecule were treated quantum mechanically by the Pariser-Parr-Pople method, while the σ electrons were described by a classical potential function.[1,2]

In 1976, Warshel and Lewitt presented a QM/MM method which can be applied more generally to complex molecular systems, and used it to study the carbonium intermediate in the cleavage of hexasaccharides, which is catalyzed by Lysozyme.[3] This was based on Lewitt’s previous work on Lysozyme, which contained the first energy minimization of an entire protein.[4,5] The paper of Warshel and Lewitt is usually seen as the birth of QM/MM.[1] The enzyme substrate complex was partitioned into a quantum mechanical and a classical region.[3] The quantum mechanical part was treated with QCFF/ALL, which is a semi-empirical method using hybrid atomic orbitals. This way, the covalent bonds between the two regions can be represented as a single orbital. The classical part is incorporated as point charges, which influence the atomic charges in the quantum region, while the atoms of the quantum system induce dipoles on the atoms in the classical region. Furthermore, the permanent dipoles of the surrounding water molecules adjust to the electric field caused by the charges of the enzyme. So all polarization effects between the different regions are included.[3]

The implementation of QM/MM was facilitated by Chandra Singh and Peter Kollman in 1986.[6,7] They simplified the energy expression by removing the polarization effects, and inserted link atoms. This allows for an easier treatment of covalent bonds connecting the quantum and the classical region.[6] Dirk Bakowies and Walter Thiel performed further investigations on the effect of polarization, and introduced a hierarchy of three different models which is still used today:[7,8] A model without any polarization is called mechanical embedding. In electrostatic embedding, the classical region polarizes the quantum system. The most sophisticated model was used in the original approach by Warshel and Lewitt, where both regions polarize each other.[8]

For their investigations, Bakowies and Thiel applied a new coupling scheme, which had been developed in the group of Keiji Morokuma.[7,9] This subtractive scheme includes a compensation of some errors that are introduced by the QM/MM partitioning. The first description of this scheme was published in 1995, where it was called IMOMM (integrated molecular orbital + molecular mechanics).[9] One year later, it was modified to IMOMO (integrated molecular orbital + molecular orbital), which combines two quantum-chemical methods instead of a quantum-chemical and a classical one.[10] In the same year, IMOMM and IMOMO were extended to ONIOM (our own n-layered integrated molecular orbital and molecular mechanics), which can combine even more than two different layers.[11] With its implementation into the Gaussian software, ONIOM became available to a large number of users.[12] In its original implementation, the subtractive scheme had the drawback that it contained no polarization effects. This was changed in 2006, when an electrostatic embedding approach for ONIOM was published.[13] Nowadays, ONIOM is one of the most successful and popular QM/MM methods.[14,15]

QM/MM approaches have been applied to a wide range of problems, most of them – but not all – concerning biological macromolecules like enzymes.[7,15] For example, they can be used for the structural refinement of enzyme substrate complexes with the interactions between enzyme and substrate described by quantum mechanics. While the contribution of specific interactions is analyzed in experimental structures (e. g. from X-ray), a refinement is even more useful for structures that are obtained by theoretical predictions (e. g. atomistic simulations or docking) and contain substrates or inhibitors not present in the experimental structure.[15] Going one step further, it is possible to investigate not only a single structure but also the mechanism of an enzyme-catalyzed reaction.[15] Examples include the catalytic mechanisms of Tryptophan Synthase (TSase) and human Fatty Acid Synthase (hFAS), which have been elucidated recently with the help of ONIOM-QM/MM.[16,17] If the mechanism of a reaction as well as the relative energies of educts, transition state, and products are known from QM/MM calculations, the enzyme can be modified and the resulting change in activation energy can be computed.[15] This way, more powerful enzymes can be designed, which are of interest for industrial applications.[15] Such a theoretical enhancement has for example been done with Dibenzothiophene desulfurization enzyme D (DszD), an enzyme that might be used for the biodesulfurization of crude oil.[18] As enzyme inhibitors are often analogs of the transition state of the catalyzed reaction, new lead structures for enzyme inhibition can be derived from the structural analysis of this reaction.[15] Furthermore, it is possible to modify known inhibitors and predict their properties, which can for example guide the development of a reversible inhibitor based on an irreversible one.[19]

2 Aim of the Work

The main objective of this work was the implementation of multiscale methods into the CAST program package, which is developed in the research group of Prof. Dr. Bernd Engels. Although the focus was on QM/MM, extensions such as QM/QM, three-layer, and multi-center methods were also implemented.

In order to provide a broader range of methods for the QM part than had been available from the energy interfaces implemented before (Gaussian, Psi4, Mopac), interfaces to some more external programs were included: DFTB$^+$ is a semi-empirical program, which is very fast and can thus be used for proof-of-concept calculations. Orca, on the other hand, features especially high-level multi-reference methods needed for the computation of open-shell systems. Furthermore, additional file formats for structure input and output have been made available in CAST. This includes the easiest possible xyz format, where CAST automatically determines covalent bonds and assigns force field atom types for most atoms that are usually present in a protein structure. Additionally, the use of pdb files was enabled, since biological macromolecules are normally provided in this format. The Umbrella Sampling task, which computes the free energy change during a reaction, was extended in order to describe reactions defined by more than one atomic distance. Two possibilities to achieve this target were implemented: The first one is a 2D version of Umbrella Sampling, where two reaction coordinates are applied. The other option uses only one reaction coordinate, but defines it as a linear combination of several atomic distances. Another modification of Umbrella Sampling that was added to CAST is PMF-IC. It allows for the correction of errors, caused by the low-level energy evaluation method that is used for the simulations, by singlepoint calculations with a high-level method. As one of the main tasks in CAST is the local optimization of structures, a specific optimizer for QM/MM was added to the corresponding energy interface. This microiteration procedure reduces the number of QM computations that are necessary during the energy minimization, which speeds up the calculation significantly. Moreover, a general optimizer from the OPT++ library was added to CAST, which allows energy minimizations with distance constraints. This is a necessary feature when performing a relaxed scan.

Last but not least, the newly implemented methods, especially the QM/MM interfaces, have been applied to the enzyme inhibitor complex consisting of rhodesain and K11777. Singlepoint computations, 2D scans, molecular dynamics simulations, and Umbrella Samplings have been performed to explore the reaction mechanism for the complex formation and evaluate the relative (free) energies along the path.

3 Theory

In general, a chemical system is described by the time-independent Schrödinger equation[20]

$$\hat{H}\Psi(R,r) = E\Psi(R,r), \tag{3.1}$$

where $\hat{H}$ is the Hamilton operator, $\Psi(R,r)$ is the wave function of the system, depending on the nuclear (R) and electronic (r) coordinates, and E is the total energy.

In most cases, the velocities of nuclei and electrons can be decoupled, because the movement of the latter is so much faster than that of the former. Therefore, the nuclei can be viewed as stationary from an electronic perspective. This is called the Born-Oppenheimer approximation. Using it, two different equations for the nuclei and the electrons are obtained:[20]

$$\hat{H}_{el}\Psi_{el}(r,R) = E_{el}(R)\Psi_{el}(r,R) \tag{3.2}$$

$$(\hat{T}_n + E_{el}(R))\Psi_n(R) = E\Psi_n(R) \tag{3.3}$$

Equation 3.2 is the electronic Schrödinger equation. Here only the movement of the electrons is taken into account, while the nuclei are stationary. The electronic wave function Ψ_{el} depends only parametrically on the nuclear coordinates R. Thus, solving this equation yields an energy $E_{el}(R)$ for every position of the nuclei. Those energies make up a multidimensional potential energy surface (PES), upon which the nuclei move. This movement is described by the nuclear Schrödinger equation (Equation 3.3).[20]

For investigating the nuclear movement of a system, the electronic Schrödinger equation has to be solved first. In case of small molecules, this can be achieved by *ab-initio* or quantum-chemical methods that are either wave function based (for example Hartree-Fock) or based on density-functional theory (DFT).[20] As those methods scale steeply with the number of basis functions, other approaches are required for bigger systems: In semi-empirical methods, parameters are introduced to reduce the number of integrals to be calculated.[20] One example of these methods is the density-functional tight-binding (DFTB), which will be described in section 3.2.[21] For very big systems, for example biomolecules, even those methods are computationally too expensive. Then the energy is evaluated by a parametric function, which is derived from a ball-and-spring model where the atoms are treated as classical particles. These approaches are called force fields or molecular mechanics.[20]

3.1 QM/MM

However, such molecular mechanics methods also have some drawbacks, as they entirely ignore the electronic structure and describe the molecule as a ball-and-spring model.[22] A very serious disadvantage is that the connectivity of the atoms (i. e. the information which balls are connected by springs) has to be given before starting the calculation. Thus, chemical reactions which are characterized by formation and breaking of bonds can't be described by a force field. Instead, a quantum-chemical or semi-empirical method has to be applied.[22]

So when for example investigating the formation of an enzyme inhibitor complex, a problem is encountered: We can't use electronic structure methods because the system is too big, but we also can't use force fields because we need to describe the rearrangement of bonds. A solution to this problem are hybrid QM/MM methods. Within them, a small part of the system where the reaction takes place is described at the quantum-chemical level of theory (QM), while the remainder is calculated by molecular mechanics (MM).[22] The principle is illustrated in Figure 3.1 where a red lightning marks the reaction site. The QM region around it is shown in yellow, the MM region in blue.

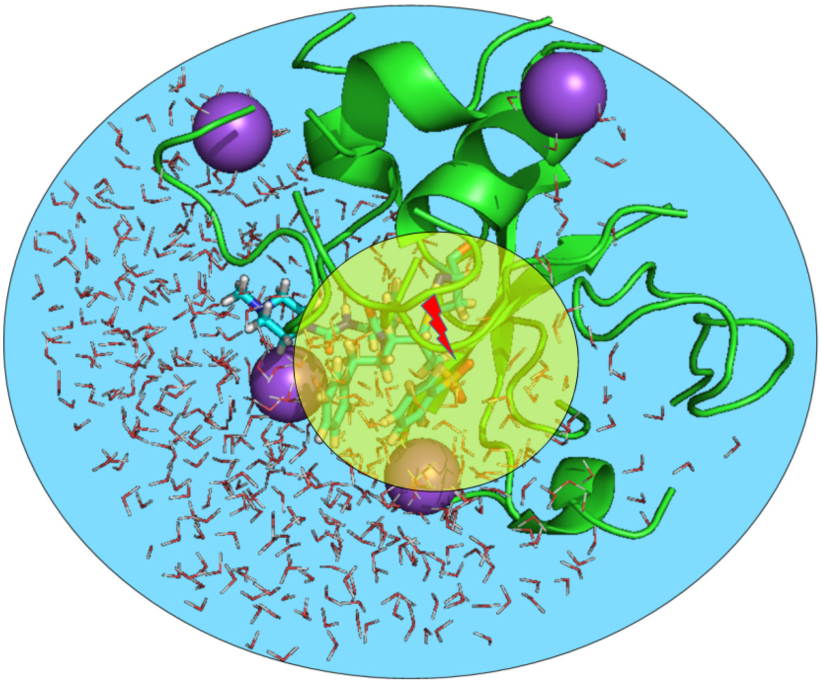


Abbildung 3.1: Illustration of QM/MM, applied on an enzyme inhibitor complex

3.1.1 Additive QM/MM

The potential energy function in QM/MM approaches consists of three parts:[22] There are interactions between atoms inside the QM region, interactions between atoms inside the MM region, and interactions between these two regions. The most straightforward way to compute the energy is just to add up those contributions. This is called the additive QM/MM scheme (also see Figure 3.2, upper part):[22]

$$E = E_{QM} + E_{MM} + E_{QM/MM} \tag{3.4}$$

If there are no bonds between QM and MM region (non-bonded QM/MM), E_{QM} and E_{MM} are obtained by just treating every region with the respective method. The description of the interactions is more complicated. In the most basic approach they are described on MM level, which means there is a van der Waals potential and a Coulomb potential between every pair of one QM and one MM atom:[22]

$$E_{QM/MM} = E_{coul} + E_{vdW} \tag{3.5}$$

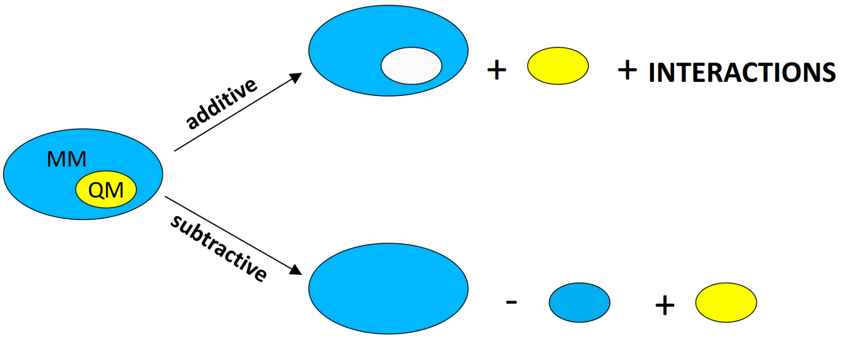


Abbildung 3.2: Illustration of QM/MM schemes

3.1.2 Subtractive QM/MM

Another possibility to calculate the QM/MM energy is first to describe the whole system (QM and MM atoms) by the MM method, then to subtract the energy of the QM system computed by the MM method, and add the energy of this system evaluated with the QM method (illustrated in Figure 3.2, lower part):[22]

$$E = E_{tot,MM} - E_{qm,MM} + E_{QM} \tag{3.6}$$

As $E_{tot,MM}$ contains all interactions between MM atoms, all interactions between QM atoms on MM level, and all interactions between QM and MM atoms on MM level, this contribution can be written as:[12]

$$E_{tot,MM} = E_{MM} + E_{qm,MM} + E_{QM/MM} \tag{3.7}$$

By substituting this into Equation 3.6, we get Equation 3.4. So the additive and the subtractive scheme are identical for non-bonded QM/MM.[12]

3.1.3 Treatment of bonds between QM and MM region

If QM system and MM system are connected by chemical bonds, the interaction energy $E_{QM/MM}$ consists not only of the Coulomb and van der Waals interactions but also contains the bonded interactions:[22]

$$E_{QM/MM} = E_{coul} + E_{vdW} + E_{bonded} \tag{3.8}$$

This bonded contribution is also described on MM level which means that for every bond, angle, and torsion where atoms of both subsystems are involved, there is a respective force field term.[22] Furthermore, the Coulomb and van der Waals potential between two atoms that are connected by one or two bonds (i. e. they are part of a bond or an angle) are not taken into account, and those potentials between atoms that are connected by four bonds (part of a torsion) are scaled down.[23] In the subtractive scheme, all of this is done automatically by the force field when evaluating $E_{tot,MM}$. In the additive scheme, however, all of those interactions have to be implemented explicitly.[22]

Another problem is that after separating QM and MM system so-called dangling bonds remain, i. e. bonds where one bonding partner is removed.[22] In the MM subsystem this usually doesn't matter because the force fields only calculate interactions between existing atoms. In quantum-chemical methods, however, the energy depends on the electronic structure. So an unpaired electron, which is created by cutting a covalent bond between QM and MM atoms, makes a big difference in comparison to this electron being a part of a bonding orbital.[22]

The common solution to this problem is the introduction of link atoms (see Figure 3.3). Then, an additional atom or atom group (in most cases hydrogen) is added to saturate the dangling bond in the QM system. These link atoms are only present in the evaluation of E_{QM} and $E_{qm,MM}$. Their position is fixed on the bonding vector between

QM and MM atoms, where the distance to the QM atom is either given or obtained by a scaling factor from the position of the MM atom.[13] No additional degrees of freedom are introduced.[22] If link atoms are included, the additive and the subtractive scheme are not identical anymore. The reason is that in the subtractive method link atoms are not only added for the calculation of E_{QM}, but also in the calculation of $E_{qm,MM}$. Therefore, there are terms that describe the interaction between link atoms and QM atoms on MM level. These interactions kind of correct the errors made by substituting the MM part of the molecule by the link atom in the QM calculation. In the additive scheme, these terms are not present.[13]

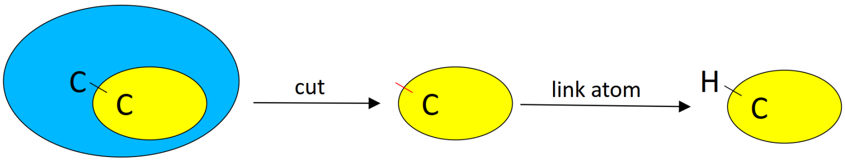


Abbildung 3.3: Illustration of the link atom approach

An alternative to link atoms is the application of localized orbitals.[22] In the localized self-consistent field (LSCF) approach, a double-occupied molecular orbital that points in the direction of the MM atom is assigned to the QM atom. This orbital is frozen during the SCF procedure in order to mimic the QM-MM bond. Another possibility is to add localized orbitals at the MM atom, a single-occupied one pointing towards the QM atom and some double-occupied and frozen ones that point in the direction of the other MM atoms. In this generalized hybrid orbital (GHO) approach, the single-occupied orbital forming the bond is optimized.[22]

A drawback of all hybrid orbital methods is that they need one or more parametrization steps to obtain the molecular orbital coefficients of the localized orbitals. Those are either performed on a different system that is used as a model or on the same system by a QM/MM calculation with a slightly bigger QM region where link atoms are applied. For this reason and because there is little improvement in accuracy, the use of link atoms is much more common.[22]

3.1.4 Treatment of the Electrostatic Interactions

Until now, all the interactions between QM and MM subsystem were treated on MM level. This **mechanical embedding** is the most basic way of handling them.[22] It can be improved by taking the partial charges from the QM calculation as charge parameters for computing the Coulomb interactions.[22]

For further enhancement, polarization effects can be included. In the **electrostatic embedding** scheme the QM region is polarized by the MM atoms. This is done by including the charges of the MM atoms into the Hamiltonian of the QM system as external point charges:[22]

$$\hat{h}_i^{mod} = \hat{h}_i - \sum_j^{\text{MM atoms}} \frac{Z_j}{|r_i - R_j|} \tag{3.9}$$

Equation 3.9 shows the modified one-electron Hamiltonian for electron i, where Z_j is the charge parameter of MM atom j, and r_i and R_j are the coordinates of the electron and the MM atom respectively. As the Hamiltonian is implemented inside the quantum chemistry software, this program has to provide an interface for the user to include external charges.[22] However, this is the case with most of the commonly used quantum chemistry software today.[24]

An issue that arises with electrostatic embedding is that the external charges do not distinguish between the atoms of the QM system but interact with all of them.[13] This may lead to an **overcounting** of some interactions as they are already included in E_{bonded}, if both the QM and the MM atom are near the boundary. Furthermore, the MM charges also interact with the link atoms, which often leads to instabilities because the distance between the link atom and the first MM atom (M1, see Figure 3.4) is even smaller than a chemical bond. Between the link atom (LA) and the M1 atom should not be any interaction because the link atom should not exist for the MM atom. But in fact the interaction is even very strong because the link atom is placed on the bond vector between Q1 and M1. There should also be no Coulomb interaction between M1 and Q1 (this is a bond term in the force field) and between M1 and Q2 (an angle contribution). The interaction between M1 and Q3 should be scaled down because they are at the both ends of a torsional angle. Some interactions of M2 and M3 are also overcounted.[13]

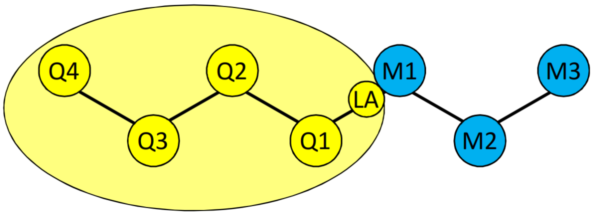


Abbildung 3.4: Illustration of over- and undercounting

One solution is to delete charges that are too close to the QM region from the external charges.[13] This solves the problem of clashing charges with the link atoms. On the other hand, one finds that while the overcounting decreases, more and more interactions are erroneously ignored. If for example the charge of atom M1 in Figure 3.4 is removed, the Coulomb interaction between M1 and Q4 is not taken into account at all. By deleting also M2 and M3, this **undercounting** also concerns interactions with QM atoms Q3, Q2 and even Q1.[13] Instead of totally ignoring those MM atoms, their charges can be scaled down or distributed to neighboring atoms.[13] It is also possible to replace the point charges by smeared-out charges that have the shape of gaussian distributions.[22] However, neither of these approaches eliminates the fundamental problem of over- and undercounting.[13]

In the additive scheme, interactions are over- and undercounted exactly as explained above. In the subtractive scheme, however, there is some favorable error compensation, similar to the link atom correction. Since the external charges are also added to the calculation of $E_{qm,MM}$, all undercounted atom pairs are neither present in E_{QM} nor in $E_{qm,MM}$ but their interactions are evaluated by $E_{tot,MM}$ on the MM level. Overcounted atom pairs, on the other hand, are present in both E_{QM} and $E_{qm,MM}$. As they are also included in $E_{tot,MM}$, the MM terms cancel out and only the interactions obtained from the QM computation remain.[13] This cancellation becomes even more accurate if the charges for the QM atoms in the two MM calculations are taken from a QM calculation instead from the force field parameters. This is due to the contributions of link atoms, which can't be equaled out by $E_{tot,MM}$, but should be nullified by the difference between E_{QM} and $E_{qm,MM}$. However, this substitution can cause problems because the charge model of QM and MM atoms in the computation of the total system is not consistent.[25]

While with electrostatic embedding the MM region polarizes the QM region, the MM atoms themselves are not polarized. This can be changed by using a **polarization embedding** scheme. There are several approaches to implement the needed polarizability into a force field:[22]

In the fluctuating charge model, the charges of the MM atoms are obtained due to the concept of electronegativity equalization. The energy of a partial charge q_A is written as a second order expansion[26]

$$E(q_A) = E_A(0) + \tilde{\chi}_A^0 q_A + \frac{1}{2} J_{AA}^0 q_A^2 \tag{3.10}$$

where J_{AA}^0 is twice the hardness of isolated atom A and $\tilde{\chi}_A^0$ is the electronegativity of that atom.[27] The electronegativity depends on the derivative of the energy U by the charge q_A:[26]

$$\tilde{\chi}_A = \frac{\partial U}{\partial q_A} \tag{3.11}$$

The energy of a system with N atoms is then[26]

$$U(q,r) = \sum_{\alpha}^{N} \left(E_\alpha(0) + \tilde{\chi}_\alpha^0 q_\alpha + \frac{1}{2} J_{\alpha\alpha}^0 q_\alpha^2 \right) + \sum_{\alpha<\beta} \Big(J_{\alpha\beta}(r_{\alpha\beta}) \cdot q_\alpha q_\beta + V(r_{\alpha\beta}) \Big) \tag{3.12}$$

where the first sum describes the energy of the charges, and the second the Coulomb and van der Waals contributions which are characterized by the terms $J_{\alpha\beta}(r_{\alpha\beta})$ and $V(r_{\alpha\beta})$ depending on the force field.[27] Now this energy is minimized by varying the charges q_α so that either the sum of all charges (intermolecular charge transfer allowed) or the sum of the charges in each molecule (no intermolecular charge transfer) is kept constant. Due to Equation 3.11, this is equivalent to equalizing the electronegativities of all atoms.[26]

An even more sophisticated polarization is reached by putting induced-dipoles on the MM atoms that can react to the electric field caused by the QM region.[24] Typically those dipoles are expressed as pairs of point charges. The main advantage over the fluctuating charge model is that the system can react not only by adjusting charges but also by adjusting the direction of the dipole. So in this model it is for example possible for benzene to be polarized by an electric field perpendicular to the molecular plane. One might also include higher order polarizibilities, but in most cases retaining only induced dipoles is a good compromise between accuracy and affordability.[24]

Another approach is the charge-on-a-spring or drude oscillator model:[28] Each atom A receives an additional particle D that carries a charge q_D, while the atomic charge q_A is reduced to $q_A - q_D$. This drude particle is connected to A by a spring with force constant k_D. Without an external electric field D oscillates around the atom, so from the outside the two particles together look like a point charge q_A. If an electric field E is applied, the drude particle reacts to it and is displaced by $d = \frac{q_D \cdot E}{k_D}$. The average induced dipole is $\mu_{ind} = \frac{q_D^2 \cdot E}{k_D}$.[28] From this last formula and the definition of the polarizibility $\alpha = \frac{\mu_{ind}}{E}$ follows:[28,29]

$$\alpha = \frac{q_D^2}{k_D} \tag{3.13}$$

The modified force field needs q_D and k_D as additional parameters. They should be chosen in a way that $\frac{q_D^2}{k_D}$ represents the polarizibility (see Equation 3.13) and the displacement d is much smaller than any interatomic distance.[28] What has to be done for polarizing the system according to an electric field is to minimize the total energy of the system by relaxing the drude particles. The total energy consists of terms totally independent from the drude particles (all interactions between the atoms), the spring energy between each atom and its drude particle, and the interactions of the drude particles with the electric field:[28]

$$U(d) = U_{atoms} + U_{spring}(d) + U_{el}(d) \tag{3.14}$$

So for each drude particle with displacement d_i the following has to be true:[28]

$$\frac{\partial U_{spring}}{\partial d_i} + \frac{\partial U_{el}}{\partial d_i} = 0 \tag{3.15}$$

Using the definition of the spring energy $\frac{1}{2}k_D d^2$ and the force on a charge in an electric field $F = q_D \cdot E$, this becomes:[28,30]

$$k_D \cdot d_i - q_{D,i} \cdot E_i = 0 \tag{3.16}$$

As E_i consists not only of the external electric field (i. e. the QM atoms) and the atom charges, but also includes all other drude charges, Equations 3.16 have to be solved in a self-consistent manner.[28]

No matter how the polarization of the MM region is reached, the procedure has to be applied at every SCF iteration of the QM calculation, which can become very computationally expensive.[22] Furthermore, polarizable force fields or modifications on unpolarizable MM potentials like described above are not broadly available.[24] In spite of polarization embedding being more accurate, the most common coupling scheme, especially for biological systems, still is electrostatic embedding.[22]

3.1.5 Comparison of Additive and Subtractive Scheme

As stated above, the additive and the subtractive scheme only lead to different results if QM and MM region are connected by covalent bonds.[25] Then, in the subtractive scheme some errors that are introduced by link atoms are corrected. However, accurate parameters for the link atoms are needed so that interactions with them cancel out between QM and MM method. The second big difference concerns the implementation: While subtractive QM/MM methods are usually based on a standard MM software, for the additive scheme the MM software has to be tailored for QM/MM. On the one hand this is a drawback because it makes the implementation more difficult, on the other hand it is an advantage as the force field can be adapted to individual wishes. For example, all of the link atom corrections mentioned above can in principle also be included into the force field implementation. In a QM/MM calculation with the additive scheme, no MM parameters are needed for QM atoms, except for those that are used for computing the interactions. In the subtractive scheme, on the contrary, a full parametrization is needed for the whole system, as it is completely included in $E_{tot,MM}$. Though many of the interactions cancel out in the end and dummy parameters can be used, most MM programs will refuse to run if any parameters are missing.[25]

So for pure QM/MM calculations additive and subtractive scheme can both be recommended, as the results depend on the specific implementation of link atom corrections and embedding scheme.[25] But only the subtractive scheme is fit for a more flexible combination of energy computation methods. Since only full saturated systems are calculated, this scheme can also be used for the combination of different quantum-chemical or semi-empirical methods.[12] Moreover, it is possible to extend it to three or more different layers or to define several QM regions in one system.[12,31]

3.2 Density-Functional Tight-Binding (DFTB)

In a QM/MM calculation, the QM part can be treated by any quantum-chemical or semi-empirical method. For most investigations in this work density-functional tight-binding (DFTB) will be used, which is a semi-empirical approach derived from density-functional theory with the approximation of tightly bound electrons.[21]

3.2.1 Density Functional Theory (DFT)

In the frame of density functional theory (DFT), the energy is written as a functional of the one-electron density $\rho(r)$[21,32]

$$E[\rho(r)] = T_e[\rho(r)] + E_{eN}[\rho(r)] + E_{ee}[\rho(r)] + E_{NN} \tag{3.17}$$

with the kinetic energy of the electrons T_e, and the interactions between electrons E_{ee}, nuclei E_{NN}, and electrons with nuclei E_{eN}.[32] If only the valence electrons are taken into account, all interactions with nuclei N are replaced by those with ions I that consist of the nuclei and the core electrons.[21] The energy E_{eN} can be viewed as the interaction of the electron density with an external potential V_{ext} that is caused by the nuclei:[32]

$$E_{eN} = \int V_{ext} \cdot \rho(r) \mathrm{d}r \tag{3.18}$$

In the Kohn-Sham theory, molecular orbitals are introduced. So the electron density is[20]

$$\rho(r) = \sum_{i}^{N_{elec}} |\chi_i(r)|^2 = \sum_{a}^{MO} f_a \cdot |\phi_a(r)|^2 \tag{3.19}$$

where f_a is the occupation number of spacial orbital ϕ_a.[21] Then the kinetic energy is approximated as sum of the kinetic energy of non-interacting electrons T_S:[20]

$$T_S = \sum_{a}^{MO} f_a \left\langle \phi_a \middle| -\frac{1}{2}\nabla^2 \middle| \phi_a \right\rangle \tag{3.20}$$

Furthermore, the electron-electron interaction is partly written as classical hartree energy which means it is calculated by the Coulomb formula:[20,21]

$$E_H = \frac{1}{2} \int \frac{\rho(r)\rho(r')}{|r' - r|} \mathrm{d}r \mathrm{d}r' = \sum_{a}^{MO} f_a \left\langle \phi_a \middle| \frac{1}{2} \int \frac{\rho(r')}{|r' - r|} \mathrm{d}r' \middle| \phi_a \right\rangle \tag{3.21}$$

As neither the non-interacting kinetic energy is exact nor the electron-electron interactions are completely represented by the hartree energy, an additional term E_{xc}, the exchange correlation functional, is inserted into the formula. It contains the errors made by replacing T_e and E_{ee} by T_S and E_H respectively:[20]

$$E_{xc}[\rho(r)] = (T_e[\rho(r)] - T_S[\rho(r)]) + (E_{ee}[\rho(r)] - E_H[\rho(r)]) \tag{3.22}$$

By including all of this into Equation 3.17, we get:[21]

$$E[\rho(r)] = \sum_{a}^{MO} f_a \left\langle \phi_a \middle| \left(-\frac{1}{2}\nabla^2 + V_{ext} + \frac{1}{2} \int \frac{\rho(r')}{|r' - r|} \mathrm{d}r' \right) \middle| \phi_a \right\rangle + E_{xc}[\rho(r)] + E_{II} \tag{3.23}$$

3.2.2 The Tight-Binding Approximation

In the tight-binding approximation, the electron density is written as a reference density, which is the superposition of atomic densities without any charge transfer, plus a density fluctuation:[21]

$$\rho(r) = \rho_0(r) + \delta\rho(r) \tag{3.24}$$

This electronic density is now substituted in Equation 3.23, resulting in

$$E[\rho] = \sum_a^{MO} f_a \left\langle \phi_a \middle| \left(-\frac{1}{2}\nabla^2 + V_{ext} + \frac{1}{2}\int \frac{\rho_0' + \delta\rho'}{|r' - r|}\mathrm{d}r' \right) \middle| \phi_a \right\rangle + E_{xc}[\rho_0 + \delta\rho] + E_{II} \tag{3.25}$$

with the notation $\rho(r) = \rho$ and $\rho(r') = \rho'$. By moving $\delta\rho$ out of the bra-kets and inserting the exchange correlation potential $V_{xc} = \frac{\partial E_{xc}[\rho]}{\partial\rho}$, this can be written as:[33]

$$\begin{aligned} E[\rho] =& \sum_a^{MO} f_a \left\langle \phi_a \middle| \left(-\frac{1}{2}\nabla^2 + V_{ext} + \frac{1}{2}\int \frac{\rho_0'}{|r' - r|}\mathrm{d}r' + V_{xc}(\rho_0) \right) \middle| \phi_a \right\rangle + E_{xc}[\rho_0 + \delta\rho] + E_{II} \\ & - \int V_{xc}(\rho_0)(\rho_0 + \delta\rho)\mathrm{d}r + \frac{1}{2}\int\int \frac{\delta\rho'(\rho_0 + \delta\rho)}{|r' - r|}\mathrm{d}r\mathrm{d}r' \end{aligned} \tag{3.26}$$

With the non-perturbed Hamiltonian $\hat{H}(\rho_0) = -\frac{1}{2}\nabla^2 + V_{ext} + \int \frac{\rho_0'}{|r'-r|}\mathrm{d}r' + V_{xc}(\rho_0)$, this becomes

$$\begin{aligned} E[\rho] =& \sum_a^{MO} f_a \left\langle \phi_a \middle| \hat{H}(\rho_0) \middle| \phi_a \right\rangle - \frac{1}{2}\int\int \frac{\rho_0'(\rho_0 + \delta\rho)}{|r' - r|}\mathrm{d}r\mathrm{d}r' + E_{xc}[\rho_0 + \delta\rho] + E_{II} \\ & - \int V_{xc}(\rho_0)(\rho_0 + \delta\rho)\mathrm{d}r + \frac{1}{2}\int\int \frac{\delta\rho'(\rho_0 + \delta\rho)}{|r' - r|}\mathrm{d}r\mathrm{d}r' \end{aligned} \tag{3.27}$$

where the double counting of the Coulomb term in the Hamiltonian has to be compensated by an additional negative term outside of the bra-kets.[33] Dividing all terms that contain $\rho + \delta\rho$ into two parts and removing those that compensate each other, we obtain:

$$\begin{aligned} E[\rho] =& \sum_a^{MO} f_a \left\langle \phi_a \middle| \hat{H}(\rho_0) \middle| \phi_a \right\rangle - \frac{1}{2}\int\int \frac{\rho_0'\rho_0}{|r' - r|}\mathrm{d}r\mathrm{d}r' + E_{xc}[\rho_0 + \delta\rho] + E_{II} \\ & - \int V_{xc}(\rho_0)\rho_0\mathrm{d}r - \int V_{xc}(\rho_0)\delta\rho\mathrm{d}r + \frac{1}{2}\int\int \frac{\delta\rho'\delta\rho}{|r' - r|}\mathrm{d}r\mathrm{d}r' \end{aligned} \tag{3.28}$$

Now the exchange energy E_{xc} is expanded in a Taylor series around ρ_0:[34]

$$\begin{aligned} E_{xc}(\rho+\delta\rho) =& E_{xc}(\rho_0) + \int \frac{\partial E_{xc}(\rho_0)}{\partial\rho}\delta\rho \mathrm{dr} + \frac{1}{2}\int\int \frac{\partial^2 E_{xc}(\rho_0)}{\partial\rho\partial\rho'}\delta\rho\delta\rho' \mathrm{dr}\mathrm{dr}' \\ &+ \frac{1}{6}\int\int\int \frac{\partial^3 E_{xc}(\rho_0)}{\partial\rho\partial\rho'\partial\rho''}\delta\rho\delta\rho'\delta\rho'' \mathrm{dr}\mathrm{dr}'\mathrm{dr}'' + \mathcal{O}(\delta\rho^4) \end{aligned} \tag{3.29}$$

For standard SCC-DFTB (also called DFTB2), this taylor series is truncated at the second order term and substituted into Equation 3.28:[33]

$$\begin{aligned} E[\rho] =& \sum_a^{MO} f_a \left\langle \phi_a \middle| \hat{H}(\rho_0) \middle| \phi_a \right\rangle + \left(E_{xc}(\rho_0) + \int \frac{\partial E_{xc}(\rho_0)}{\partial\rho}\delta\rho \mathrm{dr} + \frac{1}{2}\int\int \frac{\partial^2 E_{xc}(\rho_0)}{\partial\rho\partial\rho'}\delta\rho\delta\rho' \mathrm{dr}\mathrm{dr}' \right) \\ &- \int V_{xc}(\rho_0)\rho_0 \mathrm{dr} - \int V_{xc}(\rho_0)\delta\rho \mathrm{dr} + \frac{1}{2}\int\int \frac{\delta\rho'\delta\rho}{|r'-r|}\mathrm{dr}\mathrm{dr}' - \frac{1}{2}\int\int \frac{\rho_0'\rho_0}{|r'-r|}\mathrm{dr}\mathrm{dr}' + E_{II} \end{aligned} \tag{3.30}$$

According to the definition of the exchange potential V_{xc}, the first order term can be replaced by $\int V_{xc}(\rho_0)\delta\rho \mathrm{d}r$. After a few rearrangements the DFTB energy can be written as:

$$\begin{aligned} E[\rho] =& \sum_a^{MO} f_a \left\langle \phi_a \middle| \hat{H}(\rho_0) \middle| \phi_a \right\rangle + \frac{1}{2}\int\int \left(\frac{\partial^2 E_{xc}(\rho_0)}{\partial\rho\partial\rho'} + \frac{1}{|r'-r|} \right) \delta\rho\delta\rho' \mathrm{dr}\mathrm{dr}' \\ &+ E_{xc}(\rho_0) - \int V_{xc}(\rho_0)\delta\rho \mathrm{dr} - \frac{1}{2}\int\int \frac{\rho_0'\rho_0}{|r'-r|}\mathrm{dr}\mathrm{dr}' + E_{II} \end{aligned} \tag{3.31}$$

By defining the blue, green, and red terms as band-structure energy E_{bs}, Coulomb energy E_{coul}, and repulsive energy E_{rep} respectively, the final expression for the energy is:[21]

$$E = E_{bs} + E_{coul} + E_{rep} \tag{3.32}$$

Those three contributions will be discussed in detail in the following section.

3.2.3 Energy Contributions

Band-Structure Energy

For calculating the band-structure energy E_{bs}, the LCAO approximation is applied which means the molecular orbitals are built from atomic orbitals:[21]

$$\phi_a(r) = \sum_{\mu}^{AO} c_{\mu a} \cdot \varphi_\mu(r) \tag{3.33}$$

Since the electrons are tightly bound, a minimal basis suffices, i. e. there is one orbital for s-states, three for p-states and so on. However, atomic orbitals are too diffuse for the description of bound electrons in a molecule. For this reason, pseudo-atoms are introduced that have an additional confinement potential V_{conf} applied which is added to the Hamiltonian:[21]

$$\hat{H}(\rho_0) = -\frac{1}{2}\nabla^2 + V_{ext} + \int \frac{\rho_0'}{|r' - r|} \mathrm{d}r' + V_{xc}(\rho_0) + V_{conf}(r) \tag{3.34}$$

This confinement potential can have different forms that do not have much impact on the performance of the method. Commonly a quadratic potential is used,[21]

$$V_{conf}(r) = \left(\frac{r}{r_0}\right)^2 \tag{3.35}$$

where r_0 is a parameter which is around twice as big as the covalent radius of the atom.

By inserting 3.33 into the definition of the band-structure energy, we obtain:[21]

$$E_{bs} = \sum_{a}^{MO} f_a \sum_{\mu}^{AO} \sum_{\nu}^{AO} c^*_{\mu a} c_{\nu a} \left\langle \varphi_\mu(r) \middle| \hat{H}(\rho_0) \middle| \varphi_\nu(r) \right\rangle = \sum_{a}^{MO} f_a \sum_{\mu}^{AO} \sum_{\nu}^{AO} c^*_{\mu a} c_{\nu a} H^0_{\mu\nu} \tag{3.36}$$

In order to get the matrix elements $H^0_{\mu\nu}$, the atomic orbitals of the pseudo atom are once calculated with DFT for a given confinement potential. During the DFTB computation, they are just numbers that never change and can be saved as parameters in so-called slater-koster tables.[21]

Coulomb Energy

While the band-structure energy describes the energy of non-interacting atoms, the Coulomb energy contains all contributions that depend on the charge fluctuation $\delta\rho$, namely Coulomb and exchange interactions between the atoms.[21]

As the definition contains a double integral over all space, the first step for its calculation is to divide total space into atomic volumes $\mathscr{V}_I$:[21]

$$\int \mathrm{d}r = \sum_I \int \mathscr{V}_I \mathrm{d}r \tag{3.37}$$

Then the charge fluctuation can also be decomposed into atomic contributions

$$\delta\rho = \sum_I \Delta q_I \delta\rho_I \tag{3.38}$$

where Δq_I is the additional charge on atom I and $\delta\rho_I$ is the normalized distribution of this charge inside the atomic volume.[21]

The Coulomb energy is obtained from the sum of atom pairs IJ each of which has a Coulomb energy of:[21]

$$E^{IJ}_{coul} = \frac{1}{2}\int_{\mathscr{V}_I}\int_{\mathscr{V}_J}\left(\frac{\partial^2 E_{xc}(\rho_0)}{\partial\rho\partial\rho'} + \frac{1}{|r'-r|}\right)\Delta q_I \delta\rho_I \Delta q_J \delta\rho_J \tag{3.39}$$

Now, there is a difference whether I and J are the same atom or if they are different atoms.

If they are the same atom, 3.39 becomes:[21]

$$E^{II}_{coul} = \frac{1}{2}\Delta q_I^2 \int_{\mathscr{V}_I}\int_{\mathscr{V}_J}\left(\frac{\partial^2 E_{xc}(\rho_0)}{\partial\rho\partial\rho'} + \frac{1}{|r'-r|}\right)\delta\rho_I \delta\rho_J \tag{3.40}$$

Comparing this with Equation 3.10, we see that the formulas are analogous. Thus the double integral can be approximated by J^0_{II}, twice the hardness of atom I. This is also called Hubbard U and is approximately the difference between ionization energy and electron affinity: $U \approx IE - EA$. So for $I = J$ the Coulomb energy is calculated as:[21]

$$E^{II}_{coul} = \frac{1}{2}U_I \Delta q_I^2 \tag{3.41}$$

Since exchange interactions are local, they can be neglected if I and J are different atoms. Then Equation 3.39 can be shortened to:[21]

$$E^{IJ}_{coul} = \frac{1}{2}\int_{\mathscr{V}_I}\int_{\mathscr{V}_J}\frac{\Delta q_I \delta\rho_I \Delta q_J \delta\rho_J}{|r'-r|} = \frac{1}{2}\Delta q_I \Delta q_J \int_{\mathscr{V}_I}\int_{\mathscr{V}_J}\frac{\delta\rho_I \delta\rho_J}{|r'-r|} \tag{3.42}$$

In order to evaluate the integral, a functional form of the charge profiles $\delta\rho$ has to be specified. This can be done for example by a gaussian profile with a full width at half maximum (FWHM) of $\frac{1.329}{U}$. Then the integral is[21]

$$\int_{\mathscr{V}_I}\int_{\mathscr{V}_J}\frac{\delta\rho_I\delta\rho_J}{|r'-r|}=\frac{\mathrm{erf}(C_{IJ}R_{IJ})}{R_{IJ}} \tag{3.43}$$

with atom distance R_{IJ}, and a parameter C_{IJ} which is $\sqrt{\frac{4\ln 2}{\mathrm{FWHM}_I^2+\mathrm{FWHM}_J^2}}$. Inserting Equation 3.43 into Equation 3.42, we finally obtain:[21]

$$E_{coul}^{IJ}=\frac{1}{2}\Delta q_I\Delta q_J\cdot\frac{\mathrm{erf}(C_{IJ}R_{IJ})}{R_{IJ}} \tag{3.44}$$

For solving Equation 3.41 as well as Equation 3.44, the extra charge Δq on every atom is needed. It can be calculated as difference between the total charge of atom I and the number of valence electrons for a neutral atom:[21]

$$\Delta q_I = q_I - q_I^0 \tag{3.45}$$

The number of valence electrons q_I^0 is known from the periodic table. The total number of electrons can be expressed as the electron density in the atomic volume $\mathscr{V}_I$:[21]

$$q_I=\sum_a^{MO} f_a\int_{\mathscr{V}_I}|\phi_a(r)|^2\mathrm{d}r=\sum_a^{MO} f_a\sum_\mu^{AO}\sum_\nu^{AO} c_{\mu a}^* c_{\nu a}\int_{\mathscr{V}_I}\varphi_\mu^*(r)\varphi_\nu(r)\mathrm{d}r \tag{3.46}$$

If neither φ_μ nor φ_ν belong to atom I, the integral over $\mathscr{V}_I$ can be estimated as zero because there shouldn't be much density in this atomic volume. If on the other hand both belong to I, nearly their whole electron density is in $\mathscr{V}_I$, so the integral is approximately identical to the overlap $S_{\mu\nu}$. If one of the orbitals belongs to I and the other one does not, the integral is about half of the overlap, $\frac{1}{2}S_{\mu\nu}$. So q_I can be written as:[21]

$$q_I=\sum_a^{MO} f_a\sum_{\mu\in I}^{AO}\sum_\nu^{AO}\frac{1}{2}(c_{\mu a}^* c_{\nu a}+c_{\nu a}^* c_{\mu a})S_{\mu\nu} \tag{3.47}$$

To sum it up, the Coulomb energy in general is[21]

$$E_{coul} = \frac{1}{2}\sum_{IJ}\gamma_{IJ}(R_{IJ})\Delta q_I \Delta q_J \tag{3.48}$$

with prefactor γ

$$\gamma_{IJ}(R_{IJ}) = \begin{cases} U_I & \text{if } I = J \\ \frac{\mathrm{erf}(C_{IJ}\cdot R_{IJ})}{R_{IJ}} & \text{if } I \neq J \end{cases} \tag{3.49}$$

and extra charge Δq

$$\Delta q_I = \sum_{a}^{MO} f_a \sum_{\mu\in I}^{AO}\sum_{\nu}^{AO}\frac{1}{2}(c^*_{\mu a}c_{\nu a} + c^*_{\nu a}c_{\mu a})S_{\mu\nu} - q_I \tag{3.50}$$

Repulsive Energy

The last contribution E_{rep} consists of all interactions that do not depend on charge fluctuations.[21] It is called repulsive energy term, because it contains the ion-ion interaction E_{II} which is always repulsive. But because it also contains the exchange interactions, all the complicated physics is ignored and the energy is viewed as the sum of interactions between single atom pairs. It can then be written as

$$E_{rep} = \sum_{I<J} V^{IJ}_{rep}(R_{IJ}) \tag{3.51}$$

where functions of pair-wise interactions $V^{IJ}_{rep}(R_{IJ})$ are empirically fitted to DFT calculations.[21] For fitting, a set of structures is chosen, where each structure contains N atom pairs IJ. Its DFTB energy is

$$E_{DFTB}(R_{IJ}) = E_{bs}(R_{IJ}) + E_{coul}(R_{IJ}) + N\cdot V_{rep} + E^{const}_{rep} \tag{3.52}$$

where E^{const}_{rep} is the repulsive energy of all other atom pairs. E_{DFTB} should now be approximated by DFT, so we equate it with E_{DFT} and solve for V_{rep}:[21]

$$V_{rep}(R_{IJ}) = \frac{E_{DFT}(R_{IJ}) - E_{bs}(R_{IJ}) - E_{coul}(R_{IJ}) - E^{const}_{rep}}{N} \tag{3.53}$$

In practice, not energies are fitted but forces. The reason for this is that many applications like dynamics or geometry optimization depend on forces rather than absolute energies.

Furthermore, the biggest contribution to the DFTB energy is the band-structure part, and the short-range repulsions cannot fix inaccuracies there. So Equation 3.53 is differentiated with respect to R_{IJ}:[21]

$$V'_{rep}(R_{IJ}) = \frac{-E'_{bs}(R_{IJ}) - E'_{coul}(R_{IJ})}{N} \tag{3.54}$$

Assuming that the dataset contains only equilibrium structures, all DFT forces are zero. Now the data points (R_{IJ}, V'_{rep}) obtained from DFT calculations can be fitted to Equation 3.54. The repulsive potential is[21]

$$V_{rep}(R_{IJ}) = -\int_{R_{IJ}}^{R_{cut}} V'_{rep}(r)\mathrm{d}r \tag{3.55}$$

where R_{cut} is a parameter that describes the cutoff distance for the short-range repulsions. A good estimate of this parameter is 1.5 times the dimer distance of a homonuclear system.[21] A standard smoothing spline is chosen as fitting function. It depends on parameters λ and σ_i, where λ controls the smoothness of the curve and σ_i is a weighting factor for systems with different uncertainty σ_s and number of data points N_s. It can be calculated as $\sigma_i = \sigma_s \sqrt{N_s}$.[21]

3.2.4 Calculation of the DFTB-Energy

Substituting the results of section 3.2.3 into Equation 3.32, the following expression for the total energy is obtained:[21]

$$E = \sum_{a}^{MO} f_a \sum_{\mu}^{AO} \sum_{\nu}^{AO} c^*_{\mu a} c_{\nu a} H^0_{\mu\nu} + \frac{1}{2} \sum_{IJ} \gamma_{IJ}(R_{IJ}) \Delta q_I \Delta q_J + \sum_{I<J} V^{IJ}_{rep}(R_{IJ}) \tag{3.56}$$

The aim now is to find the MO coefficients $c_{\mu a}$ for which this energy expression is minimized. This is done by variation of the Lagrangian

$$\delta \left(E - \sum_{a}^{MO} \epsilon_a \langle \phi_a \,|\, \phi_a \rangle \right) \tag{3.57}$$

with the Lagrange multipliers ϵ_a. By solving this, the secular equations are obtained:[21]

$$\sum_{\nu}^{AO} c_{\nu a}(H_{\mu\nu} - \epsilon_a S_{\mu\nu}) = 0 \tag{3.58}$$

Energy Calculation without Self-Consistency (DFTB1)

If the charge fluctuations are ignored, i. e. the green term of Equation 3.56 is removed, the matrix elements $H_{\mu\nu}$ are identical to the parameters $H^0_{\mu\nu}$ that are obtained from DFT calculations of the pseudo atom.[33] Then the secular Equations 3.58 can be solved directly for the orbital coefficients $c_{\nu a}$ and energies ϵ_a. By inserting them back into the energy expression, it becomes:[35]

$$E_{DFTB1} = \sum_{a}^{MO} f_a \cdot \epsilon_a + \sum_{I<J} V_{rep}^{IJ}(R_{IJ}) \tag{3.59}$$

As for this energy expression it is sufficient to expand the taylor series (Equation 3.29) up to the first order term, this method is called DFTB1.[35]

Self-Consistent Charge Correction (DFTB2)

The DFTB1 scheme is quite suitable for either homonuclear or highly ionic systems. If heteronuclear covalent bonds are present, it is necessary to take into account the charge balance between the atoms of a molecule.[33] In order to do this, the Coulomb term is again added to the total energy expression. This additional term also finds its way into $H_{\mu\nu}$ of the secular equations which is then:

$$H_{\mu\nu} = H^0_{\mu\nu} + \frac{1}{2} S_{\mu\nu} \sum_{K} (\gamma_{IK} + \gamma_{JK}) \Delta q_K \tag{3.60}$$

Here I is the atom where φ_μ is located, and J the atom of φ_ν. A physical interpretation is that the charge fluctuation modifies the matrix elements $H_{\mu\nu}$ according to the electrostatic potentials of the shifted electron density.[21]

As the definition of $H_{\mu\nu}$ now contains the orbital coefficients $c_{\mu a}$ (via Δq, see Equation 3.50), the secular equations cannot be solved directly, but a self-consistent, iterative scheme has to be applied: From an initial guess for the charge fluctuation Δq, the matrix elements $H_{\mu\nu}$ can be calculated by Equation 3.60. With those, the secular equations 3.58 can be solved and coefficients $c_{\mu a}$ are obtained. They are then inserted in Equation 3.50 in order to get a new charge distribution Δq, which is the starting point for new $H_{\mu\nu}$ and so on. This is repeated until convergence is reached. At last, the converged MO coefficients can be substituted in Equation 3.56 to compute the energy.[21]

Extension of SCC-DFTB (DFTB3)

For an even more accurate description of the charge fluctuation, it is possible to include also the third order term of the taylor series 3.29. Then the energy expression becomes[34]

$$E = E_{bs} + E_{coul} + E_{rep} + E_{3rd} \tag{3.61}$$

with the third order term

$$E_{3rd} = \frac{1}{6} \int \int \int \frac{\partial^3 E_{xc}(\rho_0)}{\partial\rho\partial\rho'\partial\rho''} \delta\rho\delta\rho'\delta\rho'' \mathrm{d}\mathrm{r}\mathrm{d}\mathrm{r}'\mathrm{d}\mathrm{r}'' \tag{3.62}$$

Like the Coulomb energy in section 3.2.3, this can be decomposed into atomwise contributions:[34]

$$E_{3rd} = \frac{1}{6} \sum_{IJK} \Delta q_I \Delta q_J \Delta q_K \int_{\mathscr{V}_K} \frac{\delta}{\delta\rho''} \int_{\mathscr{V}_I} \int_{\mathscr{V}_J} \left(\frac{\partial^2 E_{xc}[\rho_0]}{\partial\rho\partial\rho'} \right) \delta\rho_I \delta\rho_J \delta\rho_K \tag{3.63}$$

The orange part of this equation is approximately γ_{IJ} from section 3.2.3. Thus, this term can be shortened to:

$$E_{3rd} = \frac{1}{6} \sum_{IJK} \Delta q_I \Delta q_J \Delta q_K \int_{\mathscr{V}_K} \frac{\partial \gamma_{IJ}}{\partial\rho''} \delta\rho_K \tag{3.64}$$

Because γ_{IJ} doesn't depend on r'', it can be differentiated with respect to q_K instead of ρ'' and be moved out of the integral. As the electron distribution $\delta\rho_K$ is normalized, the third order energy becomes:[34]

$$E_{3rd} = \frac{1}{6} \sum_{IJK} \Delta q_I \Delta q_J \Delta q_K \cdot \frac{\partial \gamma_{IJ}}{\partial q_K} \tag{3.65}$$

From the definition of the γ-function (Equation 3.49), it can be seen that only the Hubbard parameter U might depend on the charge q.[34] Let's assume that not only γ_{II} but also γ_{IJ} depends on the Hubbard parameters U_I and U_J. Then the derivative will have a value other than zero if either $K = I$ or $K = J$:

$$E_{3rd} = \frac{1}{6} \sum_{IJ} \left(\Delta q_I^2 \Delta q_J \cdot \frac{\partial \gamma_{IJ}}{\partial q_I} + \Delta q_I \Delta q_J^2 \cdot \frac{\partial \gamma_{IJ}}{\partial q_J} \right) \tag{3.66}$$

Separating terms where $I = J$ from those where $I \neq J$, we get:[34]

$$E_{3rd} = \frac{1}{3}\sum_{IJ} \Delta q_I^3 \cdot \frac{1}{2}\frac{\partial \gamma_{II}}{\partial q_I} + \frac{1}{3}\sum_{I}\sum_{J\neq I} \Delta q_I^2 \Delta q_J \cdot \frac{\partial \gamma_{IJ}}{\partial q_I} \tag{3.67}$$

As mentioned before, γ_{IJ} only depends on the charge via the Hubbard parameters. So the derivative can be written as:[34]

$$\frac{\partial \gamma_{IJ}}{\partial q_I} = \frac{\partial \gamma_{IJ}}{\partial U_I}\frac{\partial U_I}{\partial q_I} \tag{3.68}$$

While the first factor can be computed analytically, the second factor, the derivative of U_I with respect to the q_I, is given as parameter for every element.[34]

In a last step Γ is defined as

$$\Gamma_{IJ} = \begin{cases} \frac{1}{2}\frac{\partial \gamma_{II}}{\partial U_I}\frac{\partial U_I}{\partial q_I} & \text{if } I = J \\ \frac{\partial \gamma_{IJ}}{\partial U_I}\frac{\partial U_I}{\partial q_I} & \text{if } I \neq J \end{cases} \tag{3.69}$$

so we can write the third order energy contribution as:[34]

$$E_{3rd} = \frac{1}{3}\sum_{IJ} \Delta q_I^2 \Delta q_J \Gamma_{IJ} \tag{3.70}$$

After solving the Lagrangian 3.57 with the energy of Equation 3.61, the matrix elements $H_{\mu\nu}$ are[34]

$$H_{\mu\nu} = H^0_{\mu\nu} + S_{\mu\nu}\sum_K \Delta q_K \left(\frac{1}{2}(\gamma_{IK} + \gamma_{JK}) + \frac{1}{3}(\Delta q_I \Gamma_{IK} + \Delta q_J \Gamma_{JK}) + \frac{1}{6}\Delta q_K(\Gamma_{KI} + \Gamma_{KJ})\right) \tag{3.71}$$

where φ_μ is located on atom I, and φ_ν on J. As in DFTB2, the secular equations have to be solved in a self-consistent manner.[34]

A further improvement, which is in principle independent of the order of DFTB, but mostly used in combination with DFTB3, is the modification of the γ-function.[34] For a gaussian-like charge profile $\delta\rho$, this function takes the form of Equation 3.49. Often, the charge profile is also assumed as a slater function:[36]

$$\delta\rho = N \cdot \mathrm{e}^{-\tau|r-R|} \tag{3.72}$$

Then the integral γ becomes more complicated to solve. It can be written as[34,36]

$$\gamma_{IJ} = \frac{1}{R_{IJ}} - S(R_{IJ}, U_I, U_J) \tag{3.73}$$

where S is an exponentially decaying function, which ascertains that for big atomic distances only the Coulomb interaction $\frac{1}{R}$ remains. Since for $I = J$ this function should be identical to the Hubbard parameter U, the decay τ can be calculated:[36]

$$\tau = \frac{16}{5} \cdot U \tag{3.74}$$

As a physical interpretation, we can say that τ characterizes the size of the atom. If the atom is harder, τ will be bigger and thus the wave function will be more localized.[36]

For hydrogen atoms, however, the size according to Equation 3.74 is overestimated.[37] Because of this, a damping factor is introduced into γ_{IJ} if one of the atoms I and J is hydrogen. Then the so-called γ^h-function is

$$\gamma^h_{IJ} = \frac{1}{R_{IJ}} - S(R_{IJ}, U_I, U_J) \cdot \mathrm{e}^{-\left(\frac{U_I+U_J}{2}\right)^{\zeta} \cdot R^2_{IJ}} \tag{3.75}$$

with a new parameter ζ which is fitted to reproduce the water dimer.[34,37]

To sum it up, the DFTB3 extension introduces two improvements of the standard SCC-DFTB scheme:[34] The first is the extension of the taylor series to the third order term from which the method gets its name. This additional term causes the chemical hardness of an atom to depend on its charge, which is especially important for charged molecules. The second improvement is the modification of the γ-function for hydrogen atoms. It corrects the linear correlation between chemical hardness and atom size (Equation 3.74), which is only strictly valid within one row of the periodic table. This causes a much better performance in the description of hydrogen bonding. As the additional computational cost of DFTB3 in comparison to DFTB2 is negligible, the use of this method is recommended.[34]

3.2.5 Application

DFTB is two to three orders of magnitude faster than DFT, but can reach a similar performance. Thus it can be used when computational resources are limited.[37] Then it is possible to calculate much bigger systems and much longer time scales than with DFT. It is also a suitable method for gathering statistics and finding general trends by computing a large number of molecules. As it is formally derived from DFT, it can be used for developing and testing methods that should later be applied to "real" DFT.[21]

Of course, DFTB is not an *ab-initio* method, but a semi-empirical method that depends on parameterization.[21] But there is not one set of parameters that can be applied to any system and any kind of calculation.[35] In contrary, there is a whole plethora of parameter sets that are suited for different systems (e. g. organic molecules, solids, or silicons) and properties (e. g. vibrational frequencies, proton affinities, or binding energies), each of which can only be used either for SCC-DFTB or for DFTB3. If long-range correction is desired, special parameters are needed.[38] As not all elements of the periodic table are parameterized for each purpose, this limits the usage of DFTB.[35]

Independently from the parameter choice, DFTB always contains the tight-binding approximation, i. e. a minimal basis set is applied.[21] For this reason, properties which rely on a more diffuse basis, like dipole moments or polarizibility, are not very accurate.[39]

Furthermore, DFTB inherits the drawbacks from DFT, for example the bad description of charge transfer states or the lack of dispersion interactions. In many cases, solutions that have been developed to improve DFT can also be applied to DFTB, like the long-range correction or the Grimme D3 correction.[40,41]

The method that will be mostly used in this work, DFTB3 with the parameter set 3OB, performs especially well at predicting geometries, whereas in the calculation of energies larger errors in comparison to MP2 or DFT are found. For this reason, it seems promising to do conformational sampling for free energy simulations with the fast DFTB method, while single point calculations on the resulting structures are performed with a high level QM method.[42] One approach to the evaluation of free energies is Umbrella Sampling, which will be explained in the following section.

3.3 Free Energy Calculation

The driving force for chemical reactions is not the internal energy U that has been covered in sections 3.1 and 3.2, but the difference of free energy A between products and reactants.[43] Besides the enthalpy, this physical quantity also includes the entropy S, which depends on the number of possible microstates W:

$$S = \mathrm{k_B} \cdot \ln(W) \tag{3.76}$$

$\mathrm{k_B}$ is the Boltzmann constant. At a temperature T the free energy is defined as:[43]

$$A = U - T \cdot S \tag{3.77}$$

The number of microstates W and thus the entropy S are not directly computationally available. Instead, statistics are used for getting entropy and free energy. According to the Boltzmann distribution, the probability for a system to be found in state i is:[29]

$$P_i = \frac{\mathrm{e}^{-\frac{E_i}{\mathrm{k_B}T}}}{\sum_j \mathrm{e}^{-\frac{E_j}{\mathrm{k_B}T}}} \tag{3.78}$$

The denominator of Equation 3.78 is a normalization constant which is called **partition function** Z.[29] The expectation value of any quantity X is then:

$$\langle X \rangle = \sum_i P_i X_i = \frac{1}{Z} \sum_i X_i \mathrm{e}^{-\frac{E_i}{\mathrm{k_B}T}} \tag{3.79}$$

The macroscopic value of the internal energy and the entropy can be expressed as the expectation value for all an ensemble of all microstates. The inner energy becomes:[29]

$$U = \langle E \rangle = \frac{1}{Z} \sum_i E_i \mathrm{e}^{-\frac{E_i}{\mathrm{k_B}T}} \tag{3.80}$$

In an isolated system, all microstates are equally probable, so the probability for state i is $\frac{1}{W}$. Such a system is called a microcanonical ensemble.[29] With this, the entropy can be calculated:[29]

$$\begin{aligned} S &= \mathrm{k_B} \left\langle \ln \frac{1}{P} \right\rangle = \frac{\mathrm{k_B}}{Z} \sum_i \ln \left(\frac{Z}{\mathrm{e}^{-\frac{E_i}{\mathrm{k_B}T}}} \right) \cdot \mathrm{e}^{-\frac{E_i}{\mathrm{k_B}T}} \\ &= \frac{\mathrm{k_B} \ln Z}{Z} \sum_i \mathrm{e}^{-\frac{E_i}{\mathrm{k_B}T}} + \frac{1}{T}\frac{1}{Z} \sum_i E_i \mathrm{e}^{-\frac{E_i}{\mathrm{k_B}T}} \\ S &= \mathrm{k_B} \ln Z + \frac{1}{T} \langle E \rangle \end{aligned} \tag{3.81}$$

By substituting Equations 3.80 and 3.81 into Equation 3.77, another expression for the free energy is obtained:[29,44]

$$A = \langle E \rangle - T \cdot \left(\mathrm{k_B} \ln Z + \frac{1}{T} \langle E \rangle \right) = -\mathrm{k_B} T \cdot \ln Z \tag{3.82}$$

3.3.1 Potential of Mean Force (PMF)

Equation 3.82 expresses the absolute free energy of a system in dependency of the partition function Z. This partition function is calculated by adding up all possible microstates (see Equation 3.78, denominator). If those states are not discrete but continuous, i. e. when going from quantum mechanics to classical mechanics, the addition is replaced by an integration over all coordinates R:[44]

$$Z = \int \mathrm{e}^{-\frac{E(R)}{\mathrm{k_B}T}} \mathrm{d}R \tag{3.83}$$

When investigating a chemical reaction, one is normally not interested in absolute free energies, but in the relative free energy along the reaction coordinate ξ. The free energy profile $A(\xi)$ is called the Potential of Mean Force (PMF).[44] Then the partition function is substituted by the probability P to find the system at position ξ on the reaction path:[44]

$$A(\xi) = -\mathrm{k_B}T \cdot \ln P(\xi) \tag{3.84}$$

This probability $P(\xi)$ is calculated analogously to Equation 3.78 as

$$P(\xi) = \frac{\int \delta(\xi(R) - \xi) \cdot \mathrm{e}^{-\frac{E(R)}{\mathrm{k_B}T}} \mathrm{d}R}{\int \mathrm{e}^{-\frac{E(R)}{\mathrm{k_B}T}} \mathrm{d}R} \tag{3.85}$$

where the integration in the numerator is performed over all degrees of freedom except for ξ due to the Dirac delta function.[44]

3.3.2 Umbrella Sampling

Equation 3.84 provides a mathematical formula for the free energy change along a reaction path, depending on the probability distribution $P(\xi)$. What is needed now is a method to obtain $P(\xi)$, as phase space integrals cannot be calculated directly.[44] For this the **ergodic principle** is applied, which states that for a simulation of infinite time the average over time is identical to the ensemble average. This means that any statistically averaged property of the system may be estimated by a computer simulation of sufficient length, for example via molecular dynamics (MD).[44,45]

In practice, of course, it is not possible to perform a simulation of infinite length. In truncated simulations, conformations of higher energy, i. e. transition states, are usually underrepresented. As those states are also required for a realistic reaction profile, the reaction is divided into a number of windows i along the reaction coordinate ξ. Each of these windows is now sampled separately, restraining the reaction coordinate by an additional potential around the reference point ξ_i. This bias potential often has the form of a harmonic oscillator with force constant K:[44]

$$\omega_i(\xi) = \frac{1}{2}K(\xi - \xi_i) \tag{3.86}$$

From the sampling of each window, a probability distribution in the modified potential $P_{i,b}$ is obtained. In order to calculate it, the energy in Equation 3.85 is replaced by a sum of unbiased energy E_u and bias potential ω_i:[44]

$$P_{b,i}(\xi) = \frac{\int \delta(\xi(R) - \xi) \cdot \mathrm{e}^{-\frac{E_u(R)+\omega_i(\xi)}{\mathrm{k_B}T}} \mathrm{d}R}{\int \mathrm{e}^{-\frac{E_u(R)+\omega_i(\xi)}{\mathrm{k_B}T}} \mathrm{d}R} = \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \cdot \frac{\int \delta(\xi(R) - \xi) \cdot \mathrm{e}^{-\frac{E_u(R)}{\mathrm{k_B}T}} \mathrm{d}R}{\int \mathrm{e}^{-\frac{E_u(R)+\omega_i(\xi)}{\mathrm{k_B}T}} \mathrm{d}R} \tag{3.87}$$

The last step can be done because ω_i only depends on ξ and the integration is performed over all other coordinates. To obtain the unbiased distribution $P_u(\xi)$ that is needed for calculating the PMF with Equation 3.84, the relation between biased and unbiased probability is used:

$$\frac{P_{u,i}(\xi)}{P_{b,i}(\xi)} = \mathrm{e}^{+\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \cdot \frac{1}{\int \mathrm{e}^{-\frac{E_u(R)}{\mathrm{k_B}T}} \mathrm{d}R} \int \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \cdot \mathrm{e}^{-\frac{E_u(R)}{\mathrm{k_B}T}} \mathrm{d}R \tag{3.88}$$

The blue term in Equation 3.79 can be written as the expectation value $\left\langle \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \right\rangle$ of a distribution in an unbiased potential E_u. As the average is taken over all ξ, it does not depend on the reaction coordinate. It is, however, different for every window i, because the bias ω_i changes with the window. The following expression for the unbiased potential is obtained:[44]

$$P_{u,i}(\xi) = P_{b,i}(\xi) \cdot \mathrm{e}^{+\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \cdot \left\langle \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \right\rangle \tag{3.89}$$

By substituting Equation 3.89 into Equation 3.84, the PMF can be written as

$$A_i(\xi) = -\mathrm{k_B}T \cdot \ln P_{b,i}(\xi) - \omega_i(\xi) + F_i \tag{3.90}$$

with $F_i = -\mathrm{k_B}T \cdot \ln \left\langle \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \right\rangle$.[44]

Using Equation 3.90, the relative free energy $A(\xi)$ inside one window can be computed. $P_{b,i}$ is obtained by the MD simulation with a known biased potential ω_i. F_i is not a function of ξ, so it does not influence the relative values for $A(\xi)$. In order to get the whole PMF, several windows have to be combined. Then the different values for F_i in those windows must be evaluated. They cannot be obtained directly from the current window, but specific analyzing methods, for example the weighted histogram analysis method (WHAM), have to be applied on the entire set of simulations.[44]

3.3.3 Weighted Histogram Analysis Method

For the evaluation of F_i, the expectation value $\left\langle \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \right\rangle$ is required. Analogously to Equation 3.79, it can be calculated as:

$$\left\langle \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \right\rangle = \int P_u(\xi) \cdot \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \mathrm{d}\xi \tag{3.91}$$

Using Equation 3.91, F_i becomes:[44]

$$F_i = -\mathrm{k_B}T \cdot \ln \int P_u(\xi) \cdot \mathrm{e}^{-\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \mathrm{d}\xi \tag{3.92}$$

In order to solve this, the unbiased distribution P_u is needed, which is not directly available from the simulations. Instead, several distributions are obtained, one for each window i. The global distribution can be computed as a weighted average of these individual distributions,

$$P_u(\xi) = \sum_i^{windows} p_i(\xi) \cdot P_{u,i}(\xi) \tag{3.93}$$

with weighting factors $p_i(\xi)$ that are normalized, so $\sum_i^{windows} p_i(\xi) = 1$ for every ξ.[44] They are optimized by minimizing the statistical error of P_u, obtained by standard error propagation rules from the individual simulations.[45] The result of this minimization is

$$p_i(\xi) = \frac{n_i \cdot \mathrm{e}^{-\frac{\omega_i(\xi)-F_i}{\mathrm{k_B}T}}}{\sum_j^{windows} n_j \cdot \mathrm{e}^{-\frac{\omega_j(\xi)-F_j}{\mathrm{k_B}T}}} \tag{3.94}$$

where n_i is the number of MD steps in window i.[44] The unbiased distribution for one window $P_{u,i}$ can be obtained from the biased distribution, which is directly available from the simulation via Equation 3.89. With the definition of F_i, this can be written as:[46]

$$P_{u,i}(\xi) = P_{b,i}(\xi) \cdot \mathrm{e}^{+\frac{\omega_i(\xi)}{\mathrm{k_B}T}} \cdot \mathrm{e}^{-\frac{F_i}{\mathrm{k_B}T}} \tag{3.95}$$

Now, Equations 3.94 and 3.95 are substituted into Equation 3.93:[46]

$$P_u(\xi) = \sum_{i}^{windows} \frac{n_i \cdot P_{b,i}(\xi)}{\sum_j^{windows} n_j \cdot \mathrm{e}^{-\frac{\omega_j(\xi)-F_j}{\mathrm{k_B}T}}} \tag{3.96}$$

The term in the numerator, $n_i \cdot P_{b,i}(\xi)$, is the number of samples from the MD simulation with bias ω_i that are found at reaction coordinate ξ, also called $n_i(\xi)$.[47]

In practice, the conformations from the MD are separated into bins according to their position on the reaction path ξ. The unbiased probability for the system to be in bin ξ_{bin} is then:[47]

$$P_u(\xi_{bin}) = \frac{\sum_i^{windows} n_i(\xi_{bin})}{\sum_j^{windows} n_j \cdot \mathrm{e}^{-\frac{\omega_j(\xi_{bin})-F_j}{\mathrm{k_B}T}}} \tag{3.97}$$

The integral in Equation 3.92 can also be approximated by a sum over bins:[47]

$$F_i = -\mathrm{k_B}T \cdot \ln \sum_{\xi_{bin}}^{N_{bins}} P_u(\xi_{bin}) \cdot \mathrm{e}^{-\frac{\omega_i(\xi_{bin})}{\mathrm{k_B}T}} \tag{3.98}$$

3.97 and 3.98 are the two WHAM equations. As F_i is needed to get P_u (Equation 3.97) and vice versa (Equation 3.98), they have to be solved iteratively. After convergence is reached, P_u is inserted into Equation 3.84 to get the free energy profile.[44] The only information from the simulations that enters the computation is the number of samples in each bin $n_i(\xi_{bin})$. Therefore, only geometries are used for obtaining the PMF.

3.3.4 Example: Rotation of Butane

To illustrate how Umbrella Sampling works, the rotation of butane is a good example that has often been used in literature.[47,48] The reaction coordinate ξ obviously can be expressed as the torsional angle of the backbone which ranges from −180° to 180°. The path has to be separated into a certain number of windows, in this case 73, which corresponds to a window size of 5°. For each of these windows, an MD simulation has to be performed with a harmonic bias potential around the reference point ξ_i of the window (see Equation 3.86). For this example, the force constant chosen is $0.05\frac{\mathrm{kcal}}{\mathrm{mol \cdot deg^2}}$.[48] During each simulation, the current value of the torsional angle is collected in every step. Those biased distributions are plotted in Figure 3.5.

Now the distributions are separated into bins and the number of conformations in

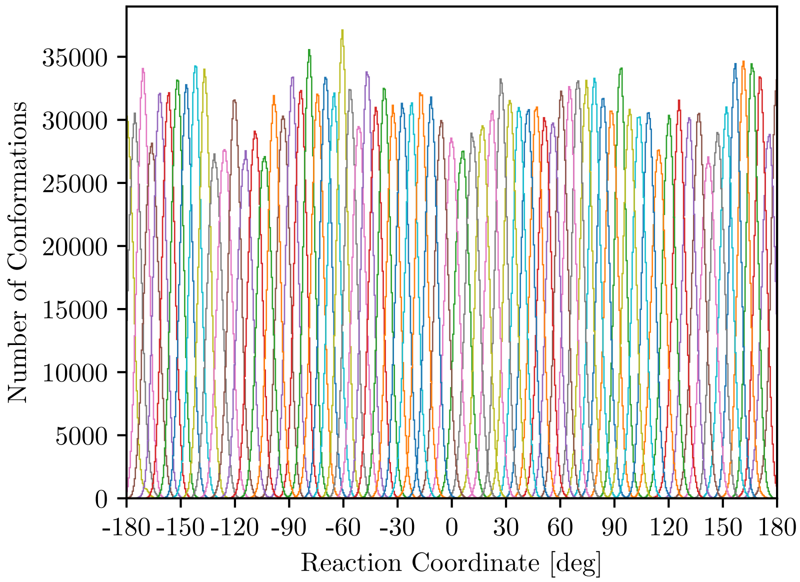
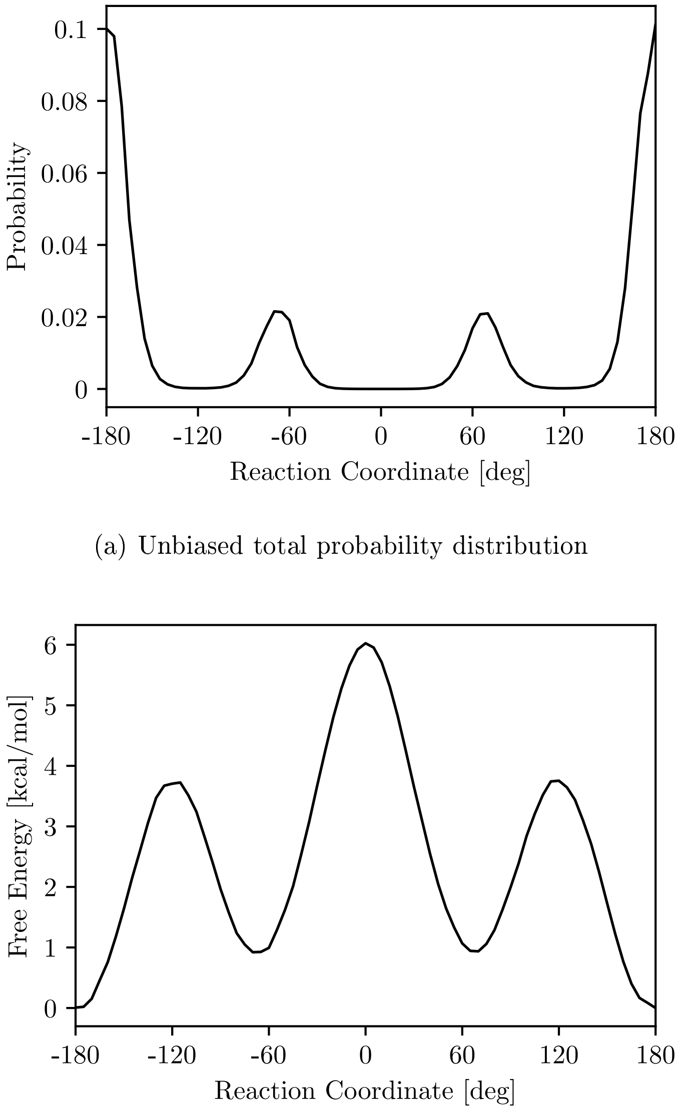


Abbildung 3.5: Biased distributions for the rotation of butane

each bin $n_i(\xi)$ is put into the WHAM equations 3.97 and 3.98 in order to obtain the unbiased total probability distribution P_u (see Figure 3.6a). For getting a continuous distribution over all ξ, the whole path has to be covered by the biased simulations, i. e. the distributions in Figure 3.5 have to overlap. If this was not the case, either additional windows would have to be applied or the force constants for the bias potentials may be modified.[49]

The last step is to take P_u and calculate the PMF according to Equation 3.84. The result is shown in Figure 3.6b. The free energy profile is in good agreement with literature.[48,50] The highest maximum is at the fully ecliptic conformation with a torsional angle of 0°. The height of this barrier is around $6\frac{\text{kcal}}{\text{mol}}$ in comparison to a literature value of $5.7\frac{\text{kcal}}{\text{mol}}$, obtained by off-path sampling.[50] Two minor barriers are found at the partially ecliptic conformations (±120°). The anti-conformation at 180° is energetically most preferred which means that it corresponds to the highest probability (see Figure 3.6a). Further local free energy minima are at the gauche conformations with torsional angles of ±60°.[48,50]

For a more intuitive explanation of WHAM, it can be said that for windows where the free energy is high (e. g. $\xi_i = 0°$) the distributions in Figure 3.5 are comparatively low and wide, because the potential forces the system to move away from reaction coordinate ξ_i which is restrained. The standard deviation for the corresponding simulations is large, so the weighting factors p_i of these windows in the total probability P_u are low. A low probability at a given point on the path corresponds to a high free energy. On the other hand, distributions at locations of low free energy (e. g. $\xi_i = -60°$) are high and narrow, which shows how a low standard deviation causes a high value of P_u in WHAM.



(a) Unbiased total probability distribution

(b) Relative Free Energy

Abbildung 3.6: Results of the Umbrella Sampling for the rotation of butane

3.3.5 Potential of Mean Force with Interpolation Correction

In Umbrella Sampling, an MD simulation has to be performed for each window.[51] In every step of each MD simulation, the energy and gradients of the system have to be evaluated. This leads to thousands or even millions of computations (in the example of section 3.3.4 with 73 windows of 500000 MD-steps each, there are 36.5 millions). Computing energy and gradients with a high-level quantum chemistry method that often is usually unfeasible. Therefore, semi-empirical or even force field methods are applied. However, the approximated description of the PES on which the simulations run might be erroneous, leading to wrong results for the PMF. This effect is especially pronounced, if the quality of the approximation varies between products and educts. For this reason, it can be reasonable to adjust the PES used for the MDs to a better method. One way to do this is the so-called interpolation correction (IC).[51]

For the correction of the surface, one starts with a number of single point calculations on structures along the reaction path. They are performed once with a high-level (HL) method and once with a low-level (LL) method, which should later be applied for the simulations. The structures used here must cover the whole reaction path and be representative of the reaction. One possibility to obtain such structures is the intrinsic reaction coordinate (IRC) search, where the reaction path is constructed starting from the transition state. The value of the correction term ΔE for each structure is the energy difference between high-level and low-level method:[51]

$$\Delta E = E_{HL} - E_{LL} \tag{3.99}$$

The values of $\Delta E(\xi)$ are interpolated using a continuous function, usually a spline under tension.[51] As spline functions are defined piecewise by polynomials, they are quite accurate for interpolation but not for extrapolation.[52] For this reason, the reaction coordinate ξ is converted to a mapping variable z, which always lies inside the finite interval $[-1, +1]$, by the transformation

$$z = \frac{2}{\pi} \arctan\left(\frac{\xi - \xi_0}{L}\right) \tag{3.100}$$

with constants ξ_0 and L, which center and scale the mapping function.[51] ξ_0 is commonly chosen as the value of ξ at the transition state, and L as a quarter of the total range of the reaction coordinate. The correction function $\Delta E(\xi)$ can now be expressed as a spline $S(z)$.[51]

During the MD simulations, the PES is modified by the spline. The energy at coordinates R is:[51]

$$E(R) = E_{LL}(R) + S(z) \tag{3.101}$$

As a consequence of this additional potential, there is an additional gradient that can be derived using the chain rule:[51]

$$\frac{\partial E(R)}{\partial R} = \frac{\partial E_{LL}(R)}{\partial R} + \frac{\partial S(z)}{\partial z} \cdot \frac{\partial z}{\partial \xi} \cdot \frac{\partial \xi}{\partial R} \tag{3.102}$$

The PMF generated from MDs with this additional gradient is called Potential of Mean Force with Interpolation Correction (PMF-IC). An example application is shown in section 5.3.3.[51]

PMF-IC with QM/MM

If the Umbrella Sampling is performed using QM/MM hybrid methods, the energy consists of a QM energy, an MM energy, and a contribution of the interactions between QM and MM atoms (see Equation 3.4):[51]

$$E = E_{QM} + E_{MM} + E_{QM/MM} \tag{3.103}$$

Usually only the QM part, that is treated by a semi-empirical method during sampling, should be corrected with a high-level quantum-chemical method. Then it is sufficient to perform the single point computations on the QM region in order to obtain the correction terms:[51]

$$\Delta E = E_{QM,HL} - E_{QM,LL} \tag{3.104}$$

In case of electrostatic embedding, external charges that correspond to the MM atoms are included in the QM calculations (see section 3.1.4). If those charges are also taken into account for the computation of ΔE, it is called perturbed interpolated correction (PMF-PIC), if E_{QM} for the correction term is calculated in gas phase, it is called unperturbed interpolation correction (PMF-UIC).[51]

4 Implementation of QM/MM

Additive and subtractive QM/MM methods as described in section 3.1 have been implemented into CAST (Conformational Search and Analysis Tool), a C++ program package developed in the research group of Prof. Dr. Bernd Engels (University of Würzburg).[53] Its features include local and global optimization routines, molecular dynamics code, and the calculation of entropies and free energies. All of these features are based on molecular energies and gradients, which can be evaluated at different levels of theory. CAST itself includes force fields, like OPLSAA[54] or CHARMM[55]. Furthermore, the software can also process the results of third-party software like Gaussian[56], Psi4[57], or Mopac[58], which provides quantum-chemical and semi-empirical methods.

In CAST, all of those energy interfaces are implemented as classes that inherit from a common base class called `interface_base`. This base class contains several pure virtual functions that have to be overridden for each interface. The most important of them are `e()` which returns the energy, and `g()` which evaluates the gradients. Gradients are stored in an object of type `Gradients_3D`. This is an alias for a container of three-dimensional vectors, so each element of it contains the gradient on the x-, y- and z-coordinate of one atom. In order to compute energy and gradients, the interface needs information about the molecular system that should be calculated, e. g. the atoms and their positions. CAST reads this kind of information from a structure input file (see also section 5.2) and saves it into an object of type `Coordinates`. The interface base class contains a pointer to such an object, so all the information is available inside the interface. The results of the calculations that are not directly returned as function value (e. g. the gradients) are also stored in this `Coordinates` object. From the outside, however, not the functions `e()` and `g()` of the energy interface are called, but the energy and gradients are computed from the `Coordinates` object. This is possible as this class contains a pointer to an object of type `interface_base`, which can point to any child class. Furthermore, the `Coordinates` class has the member functions `e()` and `g()`, which call the corresponding member functions of the energy interface. So for each new energy calculation method, a new child class of `interface_base` has to be created and the virtual functions have to be overridden. This was done for additive as well as for subtractive QM/MM.

4.1 Additive QM/MM

The additive energy interface was based on the interface `QMMM`, developed by Sara Wirsing in her bachelor thesis, which was renamed to `QMMM_A`.[59] The original implementation provided non-bonded QM/MM with electrostatic embedding, using the external software Mopac. The main task was to enable bonded connections between the QM and the MM region. Furthermore, the original interface was expanded to other quantum-chemical and semi-empirical programs that can be used as QM interfaces. Additionally, mechanical embedding was implemented into the QM/MM interface.

According to Equation 3.4, the energy in the additive QM/MM scheme is calculated as sum of the energy of the QM atoms, the energy of the MM atoms, and interactions between QM and MM atoms. So it is necessary to separate the molecular system into QM and MM atoms. This happens in the constructor of the `QMMM_A` class, where two objects of type `Coordinates` are created. One of them, called `qmc`, contains all QM atoms, the other one, `mmc`, all MM atoms. Bonds to atoms that are not present in the current `Coordinates` object (only possible in bonded QM/MM) are removed. If there are bonds between QM and MM atoms, link atoms have to be added to `qmc` (see section 4.1.1 for details). `qmc` contains a pointer to an object of the QM energy interface and `mmc` one to an object of the MM interface.

In the case of mechanical embedding (electrostatic embedding will be addressed in section 4.1.2), the energy of both systems can be obtained by simply calling the corresponding member function:

$$E_{QM} = \texttt{qmc.e()} \tag{4.1}$$

$$E_{MM} = \texttt{mmc.e()} \tag{4.2}$$

In case of non-bonded QM/MM, the interactions between QM and MM atoms consist of van der Waals and Coulomb interactions (see Equation 3.5), which are evaluated on MM level. Those interactions are calculated pairwise in force fields.[20] The Coulomb energy between two atoms A and B is

$$E_{coul}(d_{AB}) = \frac{Q_A \cdot Q_B}{\epsilon \cdot d_{AB}} \tag{4.3}$$

where d_{AB} is the distance between the atoms, Q_A and Q_B are atomic charges which are saved as force field parameters, and ϵ is the effective dielectric constant.[20] The van der Waals interaction is defined by the Lenard-Jones potential which can be written as

$$E_{vdW}(d_{AB}) = \epsilon_{AB}\left[\left(\frac{d_{AB,0}}{d_{AB}}\right)^{12} - 2\left(\frac{d_{AB,0}}{d_{AB}}\right)^{6}\right] \tag{4.4}$$

including the parameters ϵ_{AB} and $d_{AB,0}$, which correspond to the depth of the potential minimum and the minimum energy distance.[20]

Both E_{vdW} and E_{coul} (in case of mechanical embedding) are thus computed in a double loop over all QM and all MM atoms, where the pairwise energy is calculated by Equations 4.4 and 4.3. The parameters are taken from the force field parameter file of the MM interface.

The gradient G, which is the derivative of the energy with respect to the cartesian coordinates, can be separated analogously to the energy:

$$G = \frac{\partial E}{\partial R} = G_{QM} + G_{MM} + G_{QM/MM} \tag{4.5}$$

G_{QM} and G_{MM} are obtained by calling the gradient functions `qmc.g()` and `mmc.g()` respectively. However, they can not be added directly, as the gradients are not one scalar like the energy, but consist of $3N$ components, one for each cartesian coordinate of the system. In the beginning of the gradient calculation, an array of type `Gradients_3D` is created, which has the correct size to save the gradients of the whole system. Then the gradients of `qmc` are filled into the array elements corresponding to the QM atoms, and the gradients of `mmc` are assigned to the elements corresponding to the MM atoms. In order to map them, four vectors are used: `qm_indices` and `mm_indices` contain the atom indices of the QM and the MM atoms respectively. `new_indices_qm` and `new_indices_mm` have the length of the total number of atoms, where only those elements are filled that correspond to the QM (`new_indices_qm`) or MM atoms (`new_indices_mm`). They contain successive integer numbers. If the atom with index qi is a QM atom, the gradient of this atom corresponds to the element with index `new_indices_qm[qi]` of the `qmc` gradients, and the gradient for MM atom with index mi is the `new_indices_mm[mi]`'th of the `mmc` gradients.

The gradients of the interactions are computed inside the double loop, where each pairwise potential causes a gradient on both the QM and the MM atom. The formula for the gradients can be directly obtained from the Equations 4.3 and 4.4 by carrying out the differentiation.

4.1.1 Treatment of Bonded Interactions

As mentioned in section 3.1.3, two aspects have to be considered if bonds between QM and MM atoms are present: Dangling bonds need to be capped in the QM system and the interactions between QM and MM atoms must be modified to ensure that bonded interactions are also taken into account.

In CAST, the saturation of dangling bonds is done by the introduction of link atoms. They are added to `qmc` immediately when creating this object in the constructor of the QM/MM interface. For each connection between a QM atom and an MM atom, one link atom is created. It is always a hydrogen atom, although the exact force field type has to be defined by the user. As it is placed on the bonding vector between QM and MM atom at a fixed distance to the QM atom, its position can be computed as

$$R_{LA} = R_{QM} + \frac{d_{eq,LA-QM}}{d_{MM-QM}}(R_{MM} - R_{QM}) \tag{4.6}$$

where $d_{eq,LA-QM}$ is the fixed equilibrium distance between QM and MM atom which is usually taken from the force field.[60,61] If no equilibrium distance can be found in the force field parameter set (for example when providing an invalid atom type for the link atom), it is calculated as the sum of covalent radii. R_{MM} and R_{QM} are the cartesian coordinates of the MM and QM atom respectively, and d_{MM-QM} is the distance between them. The energy from `qmc.e()` is still simply inserted into Equation 3.4 as E_{QM}. The gradients, however, now also include gradients on the link atoms. As their positions depend on the positions of the QM and the MM atom, the derivative of the energy with respect to the position of the link atoms can also be expressed as a derivative with respect to the coordinates of QM atom and MM atom. This leads to an additional gradient on those two atoms.[60] Their x-component is

$$G_{QM,x} = \frac{d_{eq,LA-QM}}{d_{MM-QM}} \cdot (G_{LA} \bullet n) \cdot n_x + \left(1 - \frac{d_{eq,LA-QM}}{d_{MM-QM}}\right) \cdot G_{LA,x} \tag{4.7}$$

$$G_{MM,x} = \frac{d_{eq,LA-QM}}{d_{MM-QM}} \cdot G_{LA,x} - \frac{d_{eq,LA-QM}}{d_{MM-QM}} \cdot (G_{LA} \bullet n) \cdot n_x \tag{4.8}$$

where G_{LA} is the gradient on the link atom and n is a unit vector pointing from the QM atom in the direction of the MM atom. The subscript x marks the x-component of a vector. The y- and z-components of the gradients are treated analogously.[60]

For the calculation of bonded interactions between the QM and the MM system, atom groups forming bonds, angles, and dihedrals are searched during the initialization of the interface. In order to find system-crossing bonds, there is a loop over all QM atoms, which tests if any of their bonding partners is in the MM system. If yes, a struct of type `bonded::Bond` is created, which contains the QM as well as the MM atom. In the next step, all angles that cross the QM-MM border are found by extending every QM/MM bond by one atom to the QM side and one atom to the MM side. They are saved in structs of type `bonded::Angles`. Extension of the QM/MM angles in the same way results in the creation of QM/MM dihedrals (structs of type `bonded::dihedral`). The procedure is illustrated in Figure 4.1. The parameters necessary for evaluating the force field terms are also saved in the corresponding objects. Furthermore, each of those structs contains a function `calc_energy()`, which computes the energy and gradients on the atoms involved and is implemented analogously to the MM interface. The energy for bonds and angles is expressed as an harmonic oscillator with force constant k around an equilibrium distance d_{eq} or angle θ_{eq}:[20]

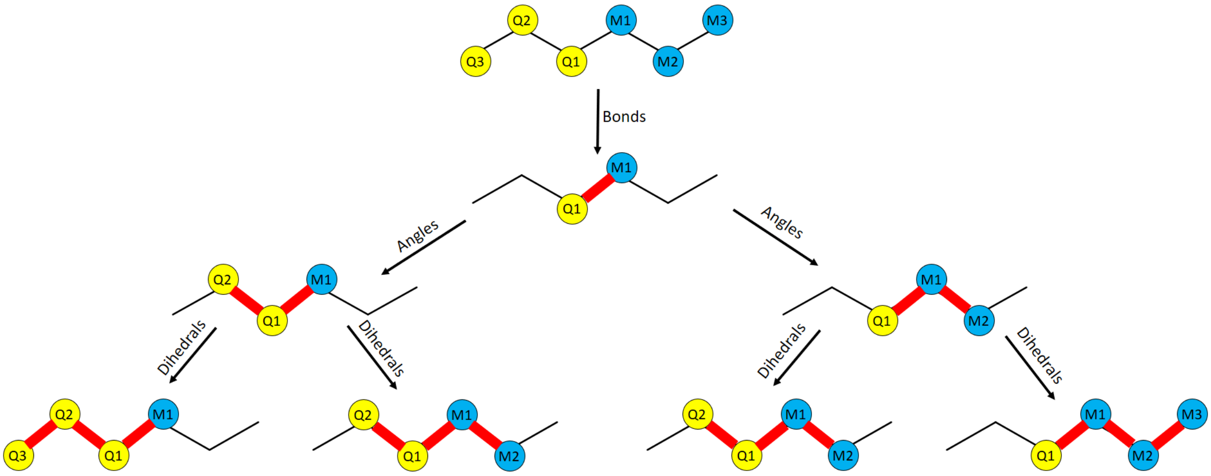


Abbildung 4.1: Procedure to find bonded interactions

$$E_{bond}(d_{AB}) = k_{AB} \cdot (R_{AB} - R_{AB,eq})^2 \tag{4.9}$$

$$E_{angle}(\theta_{ABC}) = k_{ABC} \cdot (\theta_{ABC} - \theta_{ABC,eq})^2 \tag{4.10}$$

For a dihedral angle, the energy is computed as sum of cosine functions with different periodicities and amplitudes V_1, V_2 and V_3:[20]

$$\begin{aligned} E_{dihed}(\omega_{ABCD}) =& \frac{1}{2}V_{1,ABCD} \cdot (1 + \cos(\omega_{ABCD})) + \frac{1}{2}V_{2,ABCD} \cdot (1 + \cos(2\omega_{ABCD})) \\ &+ \frac{1}{2}V_{3,ABCD} \cdot (1 + \cos(3\omega_{ABCD})) \end{aligned} \tag{4.11}$$

In the `e()` and `g()` function of the `QMMM_A` interface, the energy of each of those objects is added to the total energy, and the gradients are added to the total gradients of the corresponding atoms. Furthermore, care has to be taken to ensure the non-bonded interactions are treated specially for the atom pairs involved in bonded interactions. For this reason, a scaling factor is determined for each atom pair inside the double loops where the Coulomb and the van der Waals interactions are computed. If the two atoms are part of a `bonded::Bond` or are the outer atoms of a `bonded::Angle`, the atom pair is ignored. If the pair consists of the outer atoms of a `bonded::Dihedral`, the non-bonded interactions are scaled down by a factor saved in the force field parameters. In OPLSAA, for example, this factor is 0.5 for both van der Waals and electrostatic interactions[23]. In all other cases the interactions are calculated by Equations 4.3 and 4.4 without modification.

4.1.2 Treatment of the Electrostatic Interactions

Besides the mechanical embedding described above, the additive QM/MM interface in CAST also provides the option to use electrostatic embedding. As the Coulomb interactions are computed inside the QM interface if this option is activated (see section 3.1.4), only the van der Waals terms and bonded contributions have to be calculated explicitly:

$$E_{QM/MM} = E_{vdW} + E_{bonded} \tag{4.12}$$

So the double loop for the Coulomb interactions is inserted in an if-clause to ensure it is only executed if the user has chosen mechanical embedding.

In order to compute the electrostatic interactions in the QM interface, external charges have to be created and added to this interface. They are saved in a vector of `PointCharge`s, called `external_charges`, where `PointCharge` is a struct that stores the necessary information (especially position and charge value) for one external charge. As the charges have to be created inside the QM/MM interface, but used in the QM interface, this vector is a static member variable of the parent class `interface_base`. This means that it is shared by all objects of this class and its child classes. It is filled before calling `qmc.e()` or `qmc.g()` and cleared afterwards to ensure that the external charges are only present in the QM computation and not in the energy calculation of the MM system.

In the easiest case, when there are no bonds between QM and MM system, one charge for every MM atom is added to the vector. Its position is the position of the corresponding MM atom and its charge is the charge parameter of this atom from the force field parameter set. If there are bonds between QM and MM atoms, some of the charges are deleted for the reason of overcounting (see section 3.1.4). Those atoms that are directly bound to the QM system (M1 atoms, see Figure 3.4) are always excluded, because they crash with link atoms. This option will be called *delM1*. If neither M1 atoms nor atoms bound to them (M2 atoms) are taken into account for the creation of external charges, this

Program	Version
Gaussian	16
Psi4	1.3.2
Orca	4.4.1
Mopac	2016
DFTB$^+$	19.1

Tabelle 4.1: Software versions used in this work

will be called *delM2*. There is also the possibility to delete all charges of atoms that are bound to M2 atoms (option *delM3*). This last option is identical to the default selection *scalecharge=500* in the ONIOM implementation of the Gaussian program.[56]

The external charges have to be incorporated by the QM interface in a way that the energy as well as the gradients include the interactions between the atoms and the external charges. However, they must not include the interactions between the external charges themselves, as these are already calculated by the MM interface. Since the Coulomb interactions between QM and MM atoms also cause gradients on the MM atoms, these gradients on the external charges have to be obtained from the QM program. They are returned by a member function `get_g_ext_chg()` of the QM interface. From the QM/MM interface, they can be accessed by calling `qmc.energyinterface()->get_g_ext_chg()`.

Interfaces to five external programs have been prepared for the use with QM/MM in CAST: Those are the quantum-chemical programs Gaussian[56], Psi4[57], and Orca[62], and the semi-empirical programs Mopac[58] and DFTB$^+$[63]. The software versions used in this work are listed in Table 4.1.

In **DFTB$^+$**, external charges can be added to the computation as an external field consisting of point charges.[64] This is done by adding the following section to the input file "dftb_in.hsd":

```
ElectricField = {
  PointCharges = {
    CoordsAndCharges [Angstrom] = DirectRead {
      Records = <N_ext>
      File = 'charges.dat'
    }
  }
}
```

`<N_ext>` has to be replaced by the number of external charges. They are defined in an additional input file “charges.dat”, where every line contains the cartesian coordinates (in Ångstrom) and the charge value (in elementary charges) of one external charge. The interactions between the atoms and the external charges are automatically included in the total energy and the atom gradients. The gradients on the external charges can be read directly from the output file “results.tag”, where they are found in the section `forces_ext_charges`.[64]

Similarly straightforward is the implementation of external charges into the **Orca** interface.[65] Here, point charges are included by adding the line

```
% pointcharges pointcharges.pc"
```

to the Orca input file “orca.inp”. The point charges are then read from “pointcharges.pc”, a simple ASCII file which contains the number of external charges in the first line and the definition of the charges in the remaining lines. Each of those lines has the format “`<Q> <x> <y> <z>`” with charge Q and cartesian coordinates (x,y,z). Energy and gradients contain the interactions of the atoms with the external charges, but not the interactions between the point charges themselves. The gradients on the external charges are written into a file called “orca.pcgrad”.[65]

The interface to **Gaussian** was implemented analogously to the Amber-Gaussian interface described by Okamoto and coworkers.[66] In order to get all the necessary information, the following keywords need to be added to the `route` section of the Gaussian input file: `Charge`, `NoSymm`, and – in case of a gradient calculation – `Prop=(Field,Read)` and `Density`. `Charge` tells the program that a background charge distribution consisting of point charges is included. The external point charges are defined below the molecule specification in a section where each line of the format “`<x> <y> <z> <Q>`” corresponds to one charge. `NoSymm` prevents the system from changing its orientation, so the coordinate system remains unchanged. `Prop=(Field,Read)` requests the calculation of the electric field caused by the molecule at given positions. These positions are identical to the cartesian coordinates of the external charges and are given at the end of the input file. The keyword `Density` tells Gaussian to compute the electric field on the same level of theory as the energy. When obtaining the total energy from the logfile, special attention is necessary, because the energy printed there contains not only the QM energy and the interactions with the point charges, but also the interactions among the external charges themselves. This last contribution can be found in the logfile as “`Self interaction of the charges`” and needs to be subtracted from the total energy. The gradients on the external charges are calculated from the electric field which the QM system exerts at the given points. The electric field values can be read from the logfile, too. As they are also computed at the atomic positions, the first N entries must be ignored where N is the number of atoms in the QM calculation, i. e. QM atoms and link atoms. Then the gradients on each point charge p can be calculated as:

$$G_p = -Q_p \cdot E_{el}(R_p) \tag{4.13}$$

Q_p is the charge parameter of the corresponding MM atom and $E_{el}(R_p)$ the electric field that the QM system exerts at its position.[66]

For adding external charges to a **Psi4** calculation, first a QMMM object has to be created by inserting the command `Chrgfield = QMMM()` into the input file.[67] Then the charges are added to that object by `Chrgfield.extern.addCharge(<Q>,<x>,<y>,<z>)`. The line `Psi4.set_global_option_python('EXTERN', Chrgfield.extern)` tells Psi4 that an external charge field exists and should be taken into account. Like in Gaussian, the coordinate system has to be prevented from changing by moving the molecule to its center of mass or rotating it. This is done by adding the keywords `no_reorient` and `no_com` to the molecule definition. If gradients are required, the electric field at certain gridpoints is requested by adding the keyword `GRID_FIELD` to the one-electron property command of the Psi4 input. The gridpoints are defined in an additional file "grid.dat" as the positions of the external charges. Each line of this file contains one point in the format "`<x> <y> <z>`". Energy and atom gradients read from the output file can directly be used, since the energy does not contain the interactions between the external charges themselves as can be easily tested by comparison to a Gaussian calculation. The gradients on the point charges are calculated from the electric field values read from the output file "grid_field.dat" using Equation 4.13 .[67]

In **Mopac**, not the external charges have to be given but the additional Hamiltonian caused by them (see Equation 3.9).[68] It is written into an input file called "mol.in". This file starts with an empty line. The next line contains two numbers separated by space, the first is the number of QM atoms (without link atoms) and the second the number of link atoms. Every remaining line corresponds to one QM atom i (including link atoms) and has the format "`0 0 0 0` $\hat{h}_i^{add}$ " where the additional Hamiltonian is given as:

$$\hat{h}_i^{add} = 332 \cdot \sum_{j}^{\text{ext charges}} \frac{Q_j}{d_{ij}} \tag{4.14}$$

332 is a conversion factor needed if Q_j is given as fraction of the elementary charge (this is common for force field parameters) and the distance d_{ij} in Ångstrom. To instruct Mopac that a QM/MM calculation should be performed, the keyword `QMMM` has to be added. Then

the energy contains all electrostatic interactions between QM atoms and external charges. The gradients on the atoms, however, do not include any of those interactions at all. Instead, these contributions have to be calculated separately as derivatives of the Coulomb interactions between the MM charges (from force field parameters) and the QM charges (read from Mopac output file).[68] As the gradients on the external charges are computed in the same way, this can be done simultaneously. The pairwise Coulomb gradients are calculated in a double loop over all QM atoms and external charges, analogously to the electrostatic force field term.

4.1.3 Summary: Algorithm

As a summary of section 4.1, the algorithm for an energy and gradient calculation with the additive QM/MM interface in CAST is given here:

1. **Initializing / Updating**
    - At the start of the program, the partial `Coordinates` objects `qmc` (including link atoms) and `mmc` are created. Contributions of bonded interactions are searched and stored in structs of type `bonded::Bond`, `bonded::Angle` and `bonded::Dihedral`.
    - If several energy or gradient computations are performed during one CAST run, those objects don't have to be created from scratch. Instead, the atomic coordinates of the total system (which might change in other parts of CAST) have to be transferred to the corresponding atoms of `qmc` and `mmc`. Furthermore, the positions of the link atoms in `qmc` need to be updated.
2. **Creation of External Charges**
   If electrostatic embedding is requested, the vector `external_charges` is created from the charge parameters of the MM atoms. According to the user-defined option (*delM1*, *delM2* or *delM3*), certain MM atoms are excluded from the external point charges.
3. **Calculation of Energy and Gradients of QM system**
   The return value of `qmc.e()` or `qmc.g()` is saved in the member variable `qm_energy`. In case of electrostatic embedding, energy and gradients of `qmc` also contain the Coulomb interactions between QM and MM atoms. The gradients on the external charges (i. e. MM atoms) caused by the QM atoms are obtained from the QM interface by calling the function `get_g_ext_chg()`. They are saved in an array `g_coul_mm`, which is of type `Gradients_3D`.

4. **Deletion of External Charges**
   The vector `external_charges` is cleared to ensure that no external charges are included in the MM calculation.

5. **Calculation of Bonded Interactions between QM and MM System**
   For all structs of type `bonded::Bond`, `bonded::Angle` and `bonded::Dihedral`, energy and gradients are calculated with the respective member function. The energies are added up and stored in a variable called `bonded_energy`. The gradients are saved into the correct elements of the array `bonded_gradient` of type `Gradients_3D`, which has the size of the total system.

6. **Calculation of Van der Waals Interactions between QM and MM System**
   The van der Waals interactions between all QM and MM atoms are calculated pairwise. As for the bonded interactions, the results are saved in the variables `vdw_energy` and `vdw_gradient`.

7. **Mapping of Coulomb Gradients on External Charges to Atoms**
   In case of electrostatic embedding, the entries of `g_coul_mm` (which has the same size as the `external_charges` vector) are copied into the corresponding elements of another `Gradients_3D`-type array, `coulomb_gradient`, which has the size of the total system.

8. **Calculation of Coulomb Interactions between QM and MM System**
   In case of mechanical embedding, Coulomb energy and gradients are calculated analogously to the van der Waals interactions and saved into `coulomb_energy` and `coulomb_gradient`.

9. **Calculation of Energy and Gradients of MM system**
   The return value of `mmc.e()` or `mmc.g()` is saved into a variable called `mm_energy`. The gradients are automatically included in `mmc`.

10. **Calculation of Total Gradients**
    - `vdw_gradient`, `coulomb_gradient` and `bonded_gradient` are added up to yield the variable `new_grads`.
    - The gradient on each link atom is taken from `qmc` and partitioned to the neighboring QM and MM atom. The results are added to the corresponding elements of `new_grads`.

- The gradients of QM atoms (from `qmc`) and MM atoms (from `mmc`) are added to the corresponding elements of `new_grads`.
- The gradients `new_grads` are stored in the overall `Coordinates` object.

11. **Calculation of Total Energy**
`qm_energy`, `mm_energy`, `vdw_enery`, `bonded_energy` and `coulomb_energy` are added up and the result is returned as function value of `e()` or `g()`. In case of electrostatic embedding, the contribution of `coulomb_energy` is zero, because the Coulomb interactions are included in `qm_energy`.

4.2 Subtractive QM/MM

Because of its higher flexibility, a subtractive QM/MM method was also implemented into CAST. This was done in a new child class of `interface_base`, which is called `QMMM_S`.

As already shown in section 3.1.2, the energy of the system from a subtractive QM/MM approach is calculated by first computing the total system using the MM method ($E_{big,MM}$), then subtracting the energy of the QM region, evaluated with the MM method ($E_{small,MM}$), and finally adding the energy of the QM region, evaluated by the QM method ($E_{small,QM}$):

$$E = E_{big,MM} - E_{small,MM} + E_{small,QM} \tag{4.15}$$

These three contributions correspond to three different `Coordinates` objects in CAST. They are created in the constructor of `QMMM_S`: `qmc` contains all QM atoms (here called the *small* system) and a pointer to the QM interface, `mmc_small` contains the same atoms but a pointer to the MM interface, and `mmc_big` consists of all atoms (here called the *big* system) and also a pointer to the MM energy interface. If there are bonded connections between the QM and the MM region, the dangling bonds are saturated by link atoms in both objects that correspond to the *small* system, `qmc` and `mmc_small`. As in the additive QM/MM scheme, their position is determined by the direction of the bonding vector and the equilibrium distance to the QM atom (see Equation 4.6). In case of electrostatic embedding, external charges are also added to the computations of the two *small* `Coordinates` objects to ensure the best possible error compensation (see section 3.1.4).

Because `mmc_small` includes link atoms and is treated with the MM interface, the force field atom types of the link atoms are much more important than in the additive scheme. There, the user might assign any dummy-type, since the equilibrium distance can also be determined by covalent radii. In the subtractive scheme, however, it will not be possible to obtain any energy or gradients from `mmc_small` if the atom type of a link atom is invalid.

External Charges in Force Fields

Another important point is that the MM interface now also must be able to deal with external charges. This means it needs to include their interactions with the atoms into energy and gradients and to override the member function `get_g_ext_chg()` that returns the gradients on the external point charges. The interactions between atoms A and charges C are calculated by the Coulomb formula (see Equation 4.3), so the additional energy is:[13]

$$E_{ext} = \sum_{C} \sum_{A} \frac{Q_C \cdot Q_A}{\epsilon \cdot d_{AC}} \tag{4.16}$$

The gradient on atom A caused by the field of the external charges can be computed by differentiation of this with respect to the atomic coordinates R_A:

$$G_A = \sum_{C} \frac{Q_C \cdot Q_A}{\epsilon \cdot d_{AC}^2} \cdot \frac{R_C - R_A}{d_{AC}} \tag{4.17}$$

Analogously, the atoms cause the following gradients on the external charge C:

$$G_C = \sum_{A} \frac{Q_C \cdot Q_A}{\epsilon \cdot d_{AC}^2} \cdot \frac{R_A - R_C}{d_{AC}} \tag{4.18}$$

Combining Several Quantum-chemical Methods

While in additive QM/MM scheme the interactions between the QM and the MM system are calculated explicitly, in the subtractive scheme all energies and gradients are obtained from member functions of either the QM or the MM interface class. As those functions exist for all energy interfaces, no matter if the underlying method is a force field or an external program, the interface for the outer layer (here called MM interface) does not need to be a force field, but can also be any other method. The only condition is that it is able to deal with external point charges. Therefore, all programs mentioned in section 4.1.2 may be used.

However, there is one issue: In CAST, all user-defined options for a certain energy interface are stored in a global variable that is used for all instances of this energy interface.

So all those options are the same for mmc_small and mmc_big. This is fine for most of them, like the energy calculation method or the basisset, but not for those that depend on the system, like multiplicity or charge, as they might be different for the total and the small system. Since all systems treated in this work are singlets, no special care was taken about potentially different multiplicities. Charged systems, on the other hand, are common, so the possibility to set the charge for every system correctly is quite important. In order to take care of this, a member variable charge was added to the base class interface_base. In the interfaces applied for QM/MM (Orca, Gaussian, Psi4, DFTB$^+$ and Mopac), this member variable is set to the value of the global variable that contains the user input. In the constructor of QMMM_S, the charge is changed for the subsystem mmc_small by setting it to the value of qmc. This can be done, because those objects contain the same atoms and differ only in the energy interface that they contain a pointer to.

Another question concerns the charges to be used for the electrostatic embedding if the MM interface is not a force field and thus no force field parameters are available. In this case, atomic charges are used that are evaluated automatically by most quantum-chemical software. In order to obtain them, a virtual function charges() was added to interface_base. In the child classes of the external programs, this function was overridden in a way that it returns the charges read from the output file as a std::vector<double>. For the force field interfaces, it returns the charge parameters for all atoms. Then the MM charges can be received by calling mmc_big.charges(), no matter whether the MM interface is a force field or an external program.

Summary: Algorithm

When performing an energy or gradient calculation with the subtractive QM/MM scheme in CAST, the following algorithm is run:

1. **Initializing / Updating**
    - At the start of the program, the partial Coordinates objects qmc, mmc_small (both including link atoms) and mmc_big are created. The variable charge of mmc_small is set to the value of qmc.
    - If several energy or gradient computations are performed during one CAST run, those objects don't have to be created from scratch. Instead, the atomic coordinates of the total system (which might change in other parts of CAST) have to be transferred to the corresponding atoms of qmc, mmc_small and

mmc_big. Furthermore, the positions of the link atoms in qmc and mmc_small need to be updated.

2. **Calculation of Energy and Gradients of MM system "*big*"**
   The return value of mmc_big.e() or mmc_big.g() is saved into the member variable mm_energy_big. The gradients are stored into a local variable new_grads of type Gradients_3D. As this system contains all atoms, the resulting gradients array has the size of the total gradients and no mapping has to be done.

3. **Creation of External Charges**
   If electrostatic embedding is requested, the vector external_charges is created from the charge parameters of the MM atoms. They are taken from the charges() function of mmc_big. The QM atoms are deleted from the vector as well as those which are too close to the QM region according to the user-defined option (*delM1*, *delM2*, *delM3*).

4. **Calculation of Energy and Gradients of QM system**
   The return value of qmc.e() or qmc.g() is saved in the member variable qm_energy. The gradients for the QM atoms are added to the corresponding elements of new_grads. The gradients on the link atoms are partitioned to the neighboring QM and MM atoms (see Equations 4.7 and 4.8) and the results are also added to the corresponding elements of new_grads.

5. **Calculation of Energy and Gradients of MM system "*small*"**
   The return value of mmc_small.e() or mmc_small.g() is saved in the member variable mm_energy_small. The gradients for the QM atoms are subtracted from the corresponding elements of new_grads. The gradients on the link atoms are partitioned to the neighboring QM and MM atoms and the results are also subtracted from the corresponding elements of new_grads.

6. **Calculation of the Gradients on the External Charges**
   For qmc and mmc_small, the function get_g_ext_chg() is called, which returns the gradients that the atoms cause on the external charges. They are mapped to the correct elements of new_grads, where the gradients received from qmc are added and those from mmc_small are subtracted.

7. **Deletion of External Charges**
   The vector external_charges is cleared to ensure that no external charges are

included in the calculation for `mmc_big` if there will be more energy or gradient evaluations.

8. **Saving of Total Energy and Gradients**
   The gradient array `new_grads` is stored in the overall `Coordinates` object. The total energy is calculated from the partial energies `mm_energy_big`, `mm_energy_small` and `qm_energy` according to Equation 4.15 and returned by function `e()` or `g()` of `QMMM_S`.

4.2.1 Three-Layer Scheme

In contrast to the additive scheme, the subtractive QM/MM scheme can be extended to combine even more than two different energy evaluation methods. For example, a three-layer scheme is possible, where an additional layer, computed by a method of intermediate quality (e. g. a semi-empirical one), serves as a buffer between QM and MM system (see Figure 4.2). The total energy is then calculated as:[12]

$$E = E_{big,MM} - E_{medium,MM} + E_{medium,SE} - E_{small,SE} + E_{small,QM} \qquad (4.19)$$

As in Equation 4.15, the first subscript for each of the partial energies marks the system on which the computation is done, the second subscript marks the calculation method. MM stands for the method of the outer layer (blue in Figure 4.2), SE for that of the intermediate layer (yellow), and QM for that of the inner layer (green). Of course, this does not mean that the outer layer always needs to be treated with a force field and the intermediate one with a semi-empirical interface. Like the in the subtractive scheme, all suitable interfaces can be combined freely. Now two model systems have to be created besides the total system *big*: *medium* contains all atoms of the two inner layers (QM and SE) and *small* only those of the innermost layer (QM). Both model systems are saturated with link atoms and – in case of electrostatic embedding – external charges are included in the energy computation.

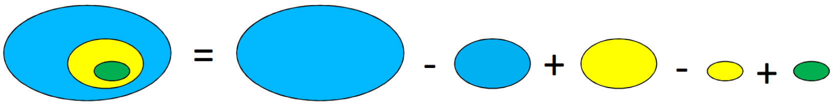


Abbildung 4.2: Illustration of three-layer scheme

In the subtractive two-layer scheme, the atom types of the link atoms are important, since they are needed for the evaluation of $E_{small,MM}$. The same is true for the atom types of the link atoms in the *medium* system of the three-layer scheme, if the MM interface is a force field, because then $E_{medium,MM}$ can only be computed with valid atom types. As it doesn't make sense to use a force field for any but the outermost layer, the atom types for the *small* system are not critically important. For this reason, only the atom types for the *medium* system have to be defined by the user, while those of the *small* system are set to a dummy-type. Their position is determined by the sum of the covalent radii.

The external point charges of the *medium* system are created from the charges of the MM atoms analogously to the two-layer scheme. For the *small* system, there are several options regarding which charges to include: The options EEx and EE are implemented in the third-party software Gaussian-ONIOM as well.[13] With EEx, no charges are added to the *small* system at all. When choosing EE, the same charges are used for both subsystems, *medium* and *small*.[13] In CAST, this option was extended to EE+. Therein, the charges of all MM atoms that are not too close to the *small* system are included, but no charges from the SE atoms are considered. The difference can be shown in Figure 4.3. When deleting only charges directly bound to the inner region (*delM1*), the charges for the *medium* system include M2 and M2'. With EE, they are identical to those used for *small*. If EE+ is chosen, atom M1 is also added to the external point charges for the *small* system, because it has three bonds distance to the nearest QM atom. On the other hand, M1' is not included, as it is too close within the frame of *delM1*. The atomic charges of the SE atoms are ignored in each of these options, assuming that the electrostatic interactions between QM and SE atoms are properly described by the *medium* system.[13] This is changed in EE+X, the last option implemented in CAST. There, the atoms of the MM and the SE layer are taken into account when creating the external point charges for the *small* system. So within the example depicted in Figure 4.3, the charges of M1, M2, M2' and S2 would be included into the computation of system *small*. While the charges of the first three atoms are taken from the MM interface, the charge of S2 is taken from the SE calculation.

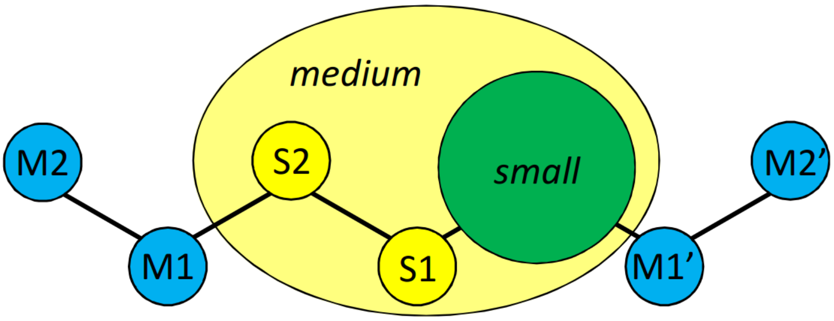


Abbildung 4.3: Illustration of electrostatic embedding in three-layer scheme

The three-layer scheme could have been implemented into the interface QMMM_S as an extension. However, for a better overview, a new interface class THREE_LAYER was added. The algorithm is nevertheless quite similar to that for the two-layer scheme described earlier:

1. **Initializing / Updating**
   - At the start of the program, five Coordinates objects corresponding to the contributions in Equation 4.19 are created: qmc_small, sec_small (both including dummy-type link atoms), sec_medium, mmc_medium (both including link atoms with user-defined atom types), and mmc_big. The variable charge of sec_small is set to the value of qmc_small and that of mmc_medium to the value of sec_medium.
   - If several energy or gradient computations are performed during one CAST run, those objects don't have to be created from scratch. Instead, the atomic coordinates of the total system (which might change in other parts of CAST) have to be transferred to the corresponding atoms of the five partial Coordinates objects. Furthermore, the positions of the link atoms in all *medium* and *small* systems need to be updated.
2. **Calculation of Energy and Gradients of MM system "*big*"**
   The return value of mmc_big.e() or mmc_big.g() is saved into the member variable mm_energy_big. The gradients are stored into a local variable new_grads of array-type Gradients_3D.
3. **Creation of External Charges for "*medium*" System**
   If electrostatic embedding is requested, the vector external_charges is created from the charges of the MM atoms. They are taken from the charges() function of mmc_big. The QM and SE atoms are deleted from the vector, as well as those which are too close to the *medium* system according to the user-defined option (*delM1*, *delM2*, *delM3*).
4. **Calculation of Energy and Gradients of SE system "*medium*"**
   The return value of sec_medium.e() or sec_medium.g() is saved in the member variable se_energy_medium. The atom gradients directly obtained by the function evaluation, as well as those received from partitioning the link atom gradients, are added to the corresponding elements of new_grads.

5. **Calculation of Energy and Gradients of MM system "*medium*"**
   The return value of `mmc_medium.e()` or `mmc_medium.g()` is saved in the member variable `mm_energy_medium`. The atom gradients directly obtained by the function evaluation, as well as those received from partitioning the link atom gradients, are subtracted from the corresponding elements of `new_grads`.

6. **Calculation of the Gradients on the External Charges due to "*medium*"**
   For `sec_medium` and `mmc_medium`, the function `get_g_ext_chg()` is called, which returns the gradients that the atoms cause on the external charges. They are mapped to the correct elements of `new_grads`, where the gradients received from `sec_medium` are added and those from `mmc_medium` are subtracted.

7. **Creation of External Charges for "*small*" System**
   If electrostatic embedding is requested, the vector `external_charges` is adjusted. According to the user-defined option, the following actions have to be taken (see also Figure 4.3):

   - EE: Nothing is done. *medium* and *small* use the same external charges.
   - EEx: The vector `external_charges` is cleared. No external charges are included in the following calculations.
   - EE+: After clearing the vector `external_charges`, the `charges()` function of `mmc_big` is called. Only those charges are added, that do not correspond to atoms in the *medium* system and are not too close to the *small* system are added.
   - EE+X: After creating the point charges for EE+, the `charges()` function of `sec_medium` is called. All charges that are not too close to the *small* system.

8. **Calculation of Energy and Gradients of QM system "*small*"**
   The return value of `qmc_small.e()` or `qmc_small.g()` is saved in the member variable `qm_energy_small`. The atom gradients directly obtained by the function evaluation, as well as those received from partitioning the link atom gradients, are added to the corresponding elements of `new_grads`.

9. **Calculation of Energy and Gradients of SE system "*small*"**
   The return value of `sec_small.e()` or `sec_small.g()` is saved in the member variable `se_energy_small`. The atom gradients directly obtained by the function evaluation, as well as those received from partitioning the link atom gradients, are subtracted from the corresponding elements of `new_grads`.

10. **Calculation of the Gradients on the External Charges due to "*small*"**
    For `qmc_small` and `sec_small`, the function `get_g_ext_chg()` is called, which returns the gradients that the atoms cause on the external charges. They are mapped to the correct elements of `new_grads`, where the gradients received from `qmc_small` are added and those from `sec_small` are subtracted.

11. **Deletion of External Charges**
    The vector `external_charges` is cleared to ensure that no external charges are included in the calculation for `mmc_big` if there will be more energy or gradient evaluations.

12. **Saving of Total Energy and Gradients**
    The gradient array `new_grads` is stored in the overall `Coordinates` object. The total energy is calculated from the partial energies according to Equation 4.19 and returned by function `e()` or `g()` of `THREE_LAYER`.

4.2.2 Multicenter QM/MM

While in the three-layer scheme more than two different computation methods are combined, it is also possible to use the subtractive QM/MM approach to apply the same computation method to more than one subsystem.[31] The principle is illustrated in Figure 4.4 by the example of two *small* systems that are treated with a high-level method (QM). The interaction between them, as well as the rest of the system, is calculated with a low-level method (MM).

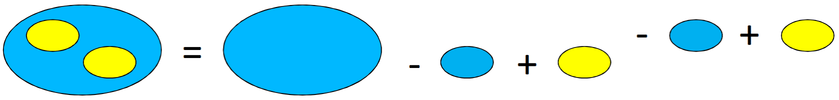


Abbildung 4.4: Illustration of Multicenter QM/MM Scheme

Generalizing this approach to N QM systems, the total energy is evaluated as:[31]

$$E = E_{big,MM} + \sum_{j}^{N} (E_{j,QM} - E_{j,MM}) \tag{4.20}$$

In CAST, this multicenter scheme was implemented as an extension to the subtractive QM/MM interface `QMMM_S` by modifying the algorithm on page 50 in the following way:

- During **Step 1** two `Coordinates` objects are created for each *small* system, one with a pointer to the QM and one with a pointer to the MM interface. They are stored in two arrays, `qmc_vec` and `mmc_small_vec`. The first array contains all systems with a pointer to the QM interface and the second all systems with a pointer to the MM interface. All of those subsystems are saturated with link atoms of user-defined force field types. In follow-up runs, all these systems including their link atoms have to be updated. As in the "normal" subtractive scheme, the total charge of the subsystems is an issue in case of QM/QM calculations. In the current implementation, it is not possible to treat *small* systems with different total charges, because the charge can only be given once for every energy interface. It is, however, possible to compute a system where the QM regions have a charge different from the total system, as the `charge` member variable of all `Coordinates` objects in `mmc_small_vec` is set to the value of the QM interface.
- **Step 3 to 7** are placed inside a loop, so they are performed once for each *small* system j. The energies `qm_energy` and `mm_energy_small` are added up in every iteration, as well as the corresponding gradients. Since the external charges are newly created for every *small* system, they contain the MM part of the molecule and the QM atoms that are not part of the current *small* region.

In systems with several reaction centers, treating all centers in separate QM systems enhances the efficiency of the calculation, as most quantum-chemical methods scale polynomial.[31] On the other hand, errors associated with the QM/MM integration are often additive, therefore the inclusion of more active sites leads to larger errors. The difference to a standard QM/MM approach, where all active sites are placed into one QM system, also increases if the centers are close to each other. In the corner case of overlapping systems, the described scheme does not work anymore, because the atoms present in both systems would be double-counted by the energy correction term in Equation 4.20.[31] There are, however, more advanced approaches to enable the calculation of such overlapping systems by the introduction of an additional reverse correction for the overlapping part.[69]

4.3 Periodic Boundary Conditions in QM/MM

Periodic boundary conditions (PBC) were implemented into the QM/MM interfaces in CAST. This was done analogously to the already existing PBC in the CAST force fields.[48] In order to understand what has been done, the general treatment of periodic boundary conditions in CAST will be explained before describing the specific implementation into the QM/MM interfaces.

4.3.1 General Introduction to Periodic Boundary Conditions

The general idea of periodic boundary conditions is that a system of infinite size can be viewed as one unit cell, replicated infinitely in all three dimensions. Then it is sufficient to perform calculations on one of these unit cells.[70] In the easiest case, the unit cell is defined as a cuboid whose edges are parallel to the x-, y- and z-axes of the coordinate system and have the lengths L_x, L_y and L_z. Any atom A with original position $R_A = (x_A, y_A, z_A)$ has images at all positions

$$R'_A = (x_A + n_x L_x, y_a + n_y L_y, z_A + n_z L_z) \tag{4.21}$$

where n_x, n_y and n_z are integers.[70]

If the user requests periodic boundary conditions in CAST, all molecules of the system are moved into one unit cell. This is done by the function `periodic_boxjump()`, which is called before each energy or gradient evaluation. If the center of mass for any molecule is outside the unit cell, the cartesian coordinates of every atom in this molecule are changed to those of another image of the atom by applying Equation 4.21. The values for n_x, n_y and n_z are chosen ensuring that the center of the molecule moves inside the unit cell. This is only a preparation step which is not strictly necessary.

The real implementation of periodic boundary conditions has to be done in the individual energy interfaces. For external programs, the PBC with the unit cell has to be defined in the input files for those programs. This option is available for DFTB$^+$, Mopac, and Gaussian, but not for Orca and Psi4.[71]

In force fields, all binding interactions can be calculated in the usual way, as one molecule is never separated into different unit cells. For all non-bonding interactions between atoms A and B, however, the distance is adjusted to the minimum image distance, i. e. the smallest possible distance of two images of A and B.[70] This is done in the function `boundary()`, which compares each component of the connecting vector between A and B with the corresponding dimension of the box size. If it is bigger than half the box size, the respective component of the box size is either added or subtracted in order to minimize the distance. The procedure is visualized in Figure 4.5 for a one-dimensional example. The box is marked in black, the two particles in yellow and green, and their distance in the original unit cell in red. If the red line is longer than half the box size, the distance between the particles is minimized by choosing the right image of the yellow particle over the left one. Then, the blue distance is used instead of the red distance for the calculation of the interactions.

This approach works well if the interactions are short-ranged in comparison to the box size. Then they can be restricted to neighboring pairs.[70] Otherwise, the situation is like the one depicted in Figure 4.5a for a repulsive interaction between two particles. The potential the yellow atom causes on the green one is plotted below. At the position of the left image it exerts the red potential, at the position of the right image it exerts the blue one. Depending on the position of the green particle, the potential corresponding to the closer image is active. When this particle goes from left to right, it moves from the red to the blue curve. In the middle, where the lines cross, the gradient changes its direction (marked by arrows). Such force discontinuities disturb energy conservation laws and thus lead to instabilities in simulations.[72]

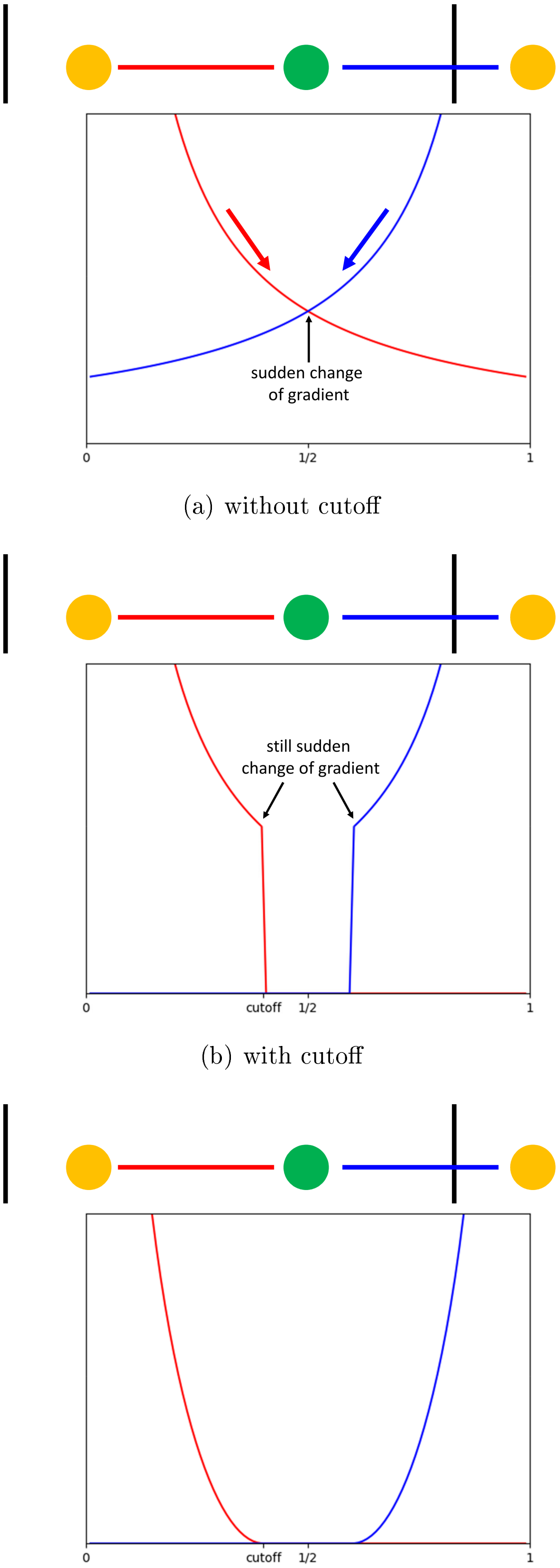
One possible solution of this problem is to introduce a cutoff for the non-bonding interactions. This means the interaction between two atoms is set to zero if their distance is bigger than a given cutoff distance c.[70] In order to have the desirable effect, this cutoff distance must be smaller than half of the box size. Figure 4.5b shows the interaction of the two particles discussed above, with a cutoff applied. The gradient does not change abruptly anymore, but there is still a discontinuity of force at the cutoff distance. This is especially an issue with Coulomb interactions, which are not decaying fast enough to describe the energy and gradient at the cutoff distance adequately by setting them to zero.[70]

To avoid those instabilities that come with an abrupt cutoff, one can scale down the interactions over some distance range.[72] Mathematically, one may introduce a switching function S which depends on the distance d. The energy contribution (e. g. the Coulomb term) is then multiplied by this switching function:[72]

$$E_{switch}(d) = S(d) \cdot E(d) \tag{4.22}$$

For a simple cutoff as described in the previous paragraph, the switching function is 1 for distances smaller than the cutoff and 0 for distances longer than the cutoff. To improve the simulation behavior, the scaling function in CAST is defined to decay gradually from 1 to 0 with increasing distance. For Coulomb interactions it reads:[72]

$$S(d) = \begin{cases} \left(1 - \left(\frac{d}{c}\right)^2\right)^2 & \text{if } d \leq c \\ 0 & \text{if } d > c \end{cases} \tag{4.23}$$



(a) without cutoff

(b) with cutoff

(c) with cutoff and switching

Abbildung 4.5: Potential between two particles with periodic boundary conditions

For van der Waals interactions, scaling is only applied between a user-defined switching distance s and the cutoff distance c:[72]

$$S(d) = \begin{cases} 1 & \text{if } d \leq s \\ \frac{(c^2-d^2)^2 \cdot (c^2+2d^2-3s^2)}{(c^2-s^2)} & \text{if } s < d \leq c \\ 0 & \text{if } d > c \end{cases} \tag{4.24}$$

Figure 4.5c shows the potential between the two particles in a periodic box with switching according to Equation 4.23. There are no discontinuities in the potential energy curve. This implies that dynamics simulations are stable and energy conservation is valid.[70]

4.3.2 Implementation of PBC into QM/MM

For the implementation of periodic boundary conditions, it is assumed that the QM system (or – in case of the three-layer interface – the *medium* system) is small in comparison to the unit cell. As a consequence, the interactions between QM systems in different cells can be neglected. Thus, the periodic boundary conditions only have to be taken into account in the MM system and for the calculation of the interactions between QM and MM atoms. For this reason, the MM interface must be able to deal with periodic boundary conditions. As already mentioned in section 4.3.1, this is the case for force fields, DFTB$^+$, Mopac, and Gaussian.

Since no interactions between the same QM system in different cells are calculated, the QM region must not be separated by the function `periodic_boxjump()`. To ascertain this, even if the QM region consists of several molecules, the function `periodic_boxjump_prep()` is called before the boxjump. This function searches all molecules and checks whether one of the atoms is a QM atom (or part of the *medium* system in the three-layer interface). All molecules, where this is the case, are combined into one container (a mock "QM molecule") for the `periodic_boxjump()` function. Therefore, they will not be separated from each other. If several QM systems are present in a multicenter QM/MM approach, avoiding separation of the QM systems becomes more complicated. In this case, such a "QM molecule" is built for every system at first. Some of these "QM molecules" might overlap if one "real" molecule takes part in several QM systems. Thus, all "QM molecules" are checked for overlapping atoms and combined if there are any, in a second step.

Inside the QM/MM interfaces, care must be taken that no periodic boundary conditions are applied to the calculation of the *small* or *medium* system. Activation of PBC is specified by a global variable `periodic` of type `bool`, which is set to `true` by the user input. In the additive scheme, it is set to `false` before starting the computation of the QM system (step 3 of the algorithm in section 4.1.3) and reset to its original value before the evaluation of the interactions (step 5). In subtractive QM/MM, periodic boundary conditions are deactivated before the computation of the *small* system (step 4 of algorithm on page 50) and set back again after clearing the external charges (step 7). For multicenter QM/MM, switching off this variable has to be done inside the loop for every *small* system, because the PBC have to be taken into account for the creation of external charges (see below). In the three-layer scheme, `periodic` is set to `false` before the computation of the *medium* system (step 4 of the algorithm on page 54), reactivated before creating the external point charges for the *small* system (step 7), and switched off again immediately after this step. At the end of the algorithm (before step 12), it is finally reset to its original value.

With these arrangements, periodic boundary conditions are automatically activated for interactions inside the MM region and for interactions between QM and MM atoms where they are computed implicitly. Those QM-MM interactions that are calculated explicitly, i. e. those in the additive scheme and the electrostatic interactions obtained through external charges, have to be considered next.

As the interactions in the additive interface are computed by the force field energy terms, periodic boundary conditions are introduced in a similar manner to the case of the force field interfaces described previously: Terms for bonds, angles, and torsions are calculated as usual. For every non-bonded pair, the distance between the atoms is adjusted to the minimum image distance, invoking the function `boundary()`. A cutoff radius, no bigger than half the box size, should be applied, and energy and gradients are scaled down with the atomic distance according to Equations 4.24 for van der Waals interactions, and 4.23 for Coulomb interactions if mechanical embedding is used.

The minimum image convention should also be applied when adding the external charges: For every charge, the image that is closest to the QM region has to be added. In order to do so, the connection vector of the charge to the center of the QM system is calculated. The external charge is moved into another unit cell, if one component of this vector is bigger than half the box size in the corresponding dimension. This is illustrated in Figure 4.6, where the original unit cell is shown in black and the QM system in yellow. Those of the original charges (blue), that are too far away from the QM system, are moved to another cell (arrows), where they are shown in green. The same procedure is applied when creating the point charges for the *medium* system in the three-layer scheme. The position of the external charges for the *small* system depends on the user-defined option: While with EE the charges of the *medium* system remain unchanged around the *medium* system, with EE+ and EE+X they are newly created and positioned around the *small* system. For this reason, EE might not be a good choice to use with periodic boundary conditions if the centers of those two regions are not close to each other.

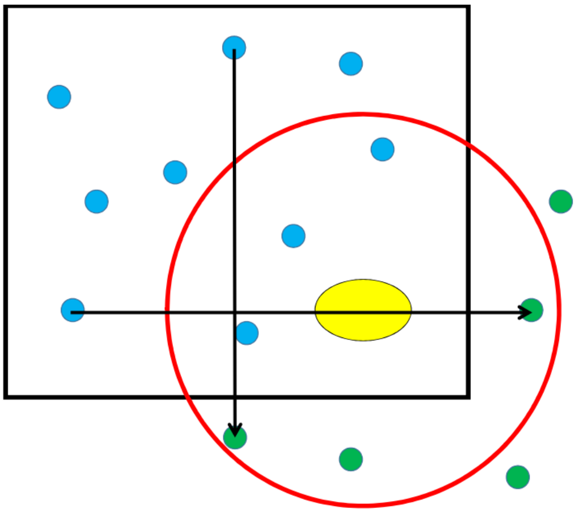


Abbildung 4.6: Distribution of external charges in a QM/MM calculation with periodic boundaries

Analogously to the interactions in force fields, a cutoff should be applied to the external charges. It should be at most half the value of the box size in the shortest dimension in order to avoid force discontinuities. This cutoff distance is marked red in Figure 4.6. All charges that are outside of the circle should not be taken into account. However, Coulomb interactions do not decay fast enough for an abrupt cutoff to yield stable simulations (see also section 4.3.1). Instead, the interactions should be scaled down continuously. This is done by multiplication of each external charge Q by the switching function:

$$Q_{shift}(d) = Q \cdot S(d) \tag{4.25}$$

The switching function $S(d)$ is identical to the one used for the Coulomb interactions in the force fields (Equation 4.23). The distance d is calculated with respect to the center of the QM region, which could be intuitively chosen to be the geometrical center of all

QM atoms. However, to facilitate the calculations, it is always defined as the position of one QM atom in CAST. The atom that is nearest to the geometrical center is chosen by default, but the user can also specify another one.

By delivering the shifted charges Q_{shift} instead of Q to the external program, the Coulomb energy is automatically scaled down correctly by the switching factor S. For the gradients it is not that simple, because the scaling factor changes with the distance d and therefore depends on the atomic coordinates. The QM interface only receives the (scaled) charges and their positions, and treats their charge values as fixed. Therefore, gradient contributions caused by variation of charges with their positions must be calculated separately:

It is assumed that the energy of an external charge C can be expressed as sum of Coulomb interactions with all QM atoms A:

$$E_{ext,C} = \sum_A \frac{Q_A \cdot Q_{C,shift}(d)}{\epsilon \cdot |R_A - R_C|} = \sum_A \frac{Q_A \cdot Q_C \cdot S(d)}{\epsilon \cdot |R_A - R_C|} \tag{4.26}$$

As d is the distance between the external charge and the QM center atom B, the switching function depends on the coordinates of those two particles:

$$E_{ext,C} = \sum_A \frac{Q_A \cdot Q_C \cdot S(R_B, R_C)}{\epsilon \cdot |R_A - R_C|} \tag{4.27}$$

Now the derivative of this energy with respect to the coordinates R is computed. As Q_A, Q_C, and ϵ are constants, the gradients can be written as:

$$\frac{\partial E_{ext,C}}{\partial R} = \sum_A \frac{Q_A \cdot Q_C}{\epsilon} \cdot \frac{\partial}{\partial R} \frac{S(R_B, R_C)}{|R_A - R_C|} \tag{4.28}$$

By applying the product rule to the last part, we get:

$$\frac{\partial E_{ext,C}}{\partial R} = S(R_B, R_C) \cdot \left(\frac{\partial}{\partial R} \sum_A \frac{Q_A \cdot Q_C}{\epsilon \cdot |R_A - R_C|} \right) + \left(\sum_A \frac{Q_A \cdot Q_C}{\epsilon \cdot |R_A - R_C|} \right) \cdot \left(\frac{\partial}{\partial R} S(R_B, R_C) \right) \tag{4.29}$$

The first addend corresponds to the gradients computed by the QM program. Only the second addend (marked in blue) has to be explicitly evaluated by the QM/MM interfaces. Since the derivative of S with respect to all other coordinates except R_B and R_C is zero, there is only an additional gradient on the central atom B and the external charge C. In order to obtain this gradient, the derivatives of Equation 4.23 with respect to R_B and R_C

have to be computed. With the definition $d = |R_B - R_C|$ they are:

$$\begin{aligned} \frac{\partial S(R_B, R_C)}{\partial R_C} &= \frac{4(R_B - R_C) \cdot \sqrt{S(R_B, R_C)}}{c^2} \\ \frac{\partial S(R_B, R_C)}{\partial R_B} &= -\frac{4(R_B - R_C) \cdot \sqrt{S(R_B, R_C)}}{c^2} \end{aligned} \tag{4.30}$$

For every external charge, there is an additional gradient on the corresponding MM atom with magnitude

$$G_{C,switch} = \frac{4(R_B - R_C) \cdot \sqrt{S(R_B, R_C)}}{c^2} \cdot \left(\sum_A \frac{Q_A \cdot Q_C}{\epsilon \cdot |R_A - R_C|} \right) \tag{4.31}$$

and one on the QM atom that defines the center:

$$G_{B,switch} = -\frac{4(R_B - R_C) \cdot \sqrt{S(R_B, R_C)}}{c^2} \cdot \left(\sum_A \frac{Q_A \cdot Q_C}{\epsilon \cdot |R_A - R_C|} \right) \tag{4.32}$$

These gradients are exact within the assumption that the interaction of an atomic system with the external charges is correctly described as a sum of Coulomb interactions with atomic point charges. As shown in sections 4.1.2 and 4.2, this is valid for the force fields and the Mopac interface, but not for Gaussian and Psi4 where the external charges interact with the electric field that is caused by a delocalized charge distribution. For this reason, there might be some artefacts when performing a QM/MM simulation with these interfaces if the cutoff is too small. Test simulations with a very small cutoff of 4 Å showed that those artefacts are also present using Orca but not when using DFTB$^+$. In the semi-empirical program DFTB$^+$, the external charges seem to interact with point charges, whereas in the quantum-chemical program Orca, there is a delocalized charge cloud. However, if a reasonable cutoff is chosen, the energy fluctuations caused by those artefacts are expected to be negligible.

5 Further Implementation

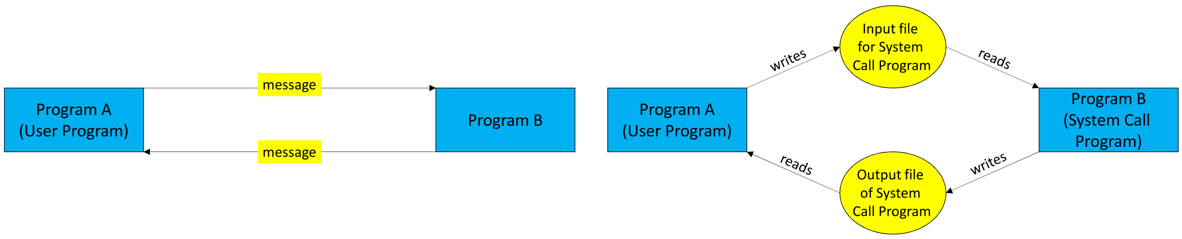
Besides the implementation of the QM/MM interfaces described in chapter 4, several extensions were added to different parts of the CAST program package to enhance user experience or to provide new functionality. Most of those extensions are especially helpful when using them in combination with the QM/MM methods.

5.1 External Energy Interfaces

As in QM/MM two or three different energy interfaces are combined, the applicability can be improved by having a bigger choice of suitable energy interfaces. For this reason, interfaces to two more external programs were implemented into CAST: Orca is a multi-purpose quantum-chemical software package that features a lot of methods, including multireference methods like CAS-SCF.[62] DFTB$^+$ is an implementation of density-functional tight-binding including extensions like DFTB3, dispersion correction, or range separation.[64] Both programs are designed to be used within QM/MM. Thus, the infrastructure necessary for inclusion of external charges is provided, which is the prerequisite for their integration into CAST-QM/MM (also see section 4.1.2).[64,73]

Communication between two programs can be achieved either by a system call or by means of a message passing interface (MPI). When using MPI, several processes are created at the start. The programs can run simultaneously and communicate directly with each other (Figure 5.1a). An example of an MPI is the Terachem interface included in CAST.[53] DFTB$^+$ and Orca were both implemented as system call interfaces. This means that the operating system (OS) is used to call one program from the other. The syntax for such a system call in C++ depends on the OS. On Unix-like platforms (Linux) the function `system()` from the standard library is used, while on Windows the function `CreateProgress()` from the Win32 Application Programming Interface (API) may be applied for the same purpose.[74,75] Unlike in an MPI, the two programs are not by default executed simultaneously. Instead, the calling program is interrupted while the other program runs. For this reason, there cannot be any direct communication between caller and callee, but the two programs need to communicate using file input and output (Figure 5.1b).

The reading and writing of files on the hard drive is, of course, slower than sending the information via MPI between parallel processes. However, not every program provides the feature to be accessed via MPI, whereas system calls can always be used. In case of the programs included here, DFTB$^+$ and Orca, the time needed for the calculations is much



(a) MPI

(b) System call

Abbildung 5.1: Different ways of communication between two programs

longer than that of file input and output. Therefore, the performance loss is probably negligible.

5.1.1 Implementation Details

As both the DFTB$^+$ and the Orca interface are system call interfaces, their implementations are quite similar. For each of the two, a new child class of `interface_base` was created. The one for DFTB$^+$ is called `dftb::sysCallInterface`, the one for Orca is called `orca::sysCallInterface`. Some virtual functions of the parent class need to be overridden. The most important of these virtual functions, `e()` and `g()`, which calculate the energy and gradients, were already explained in chapter 4. Futhermore, there is a function called `h()` which computes the hessian matrix, and another one, `o()` which performs a local optimization within the current energy interface (see also section 5.4). Each of those four functions executes the following steps:

At first, the input file for the external program is written. For DFTB$^+$, it is called "dftb_in.hsd", and for Orca "orca.inp". This is done in a function called `write_inputfile()`, which gets passed an integer as a parameter that corresponds to the calculation mode. "0" requests only the energy, "1" also the gradients, "2" the hessian calculation, and "3" an optimization procedure. Information needed for successful execution includes the cartesian coordinates of the atoms which are taken from the `Coordinates` object, the user options for the external program which will be described in detail in sections 5.1.2 and 5.1.3, and – if periodic boundary conditions are applied – the size of the unit cell. If point charges are present in the vector `external_charges`, they are written into the files "charges.dat" or "pointcharges.pc" (see section 4.1.2).

After the creation of those files, the program is executed via a system call, where the standard output of the target program is directed into a file. The commands are:

```
<path to dftb+> > output_dftb.txt  # for DFTB+
<path to orca>  > output_orca.txt  # for Orca
```

The paths of the third-party binaries have to be provided by the user in the CAST input file.

After the program has finished successfully, CAST reads the necessary information from the output files. DFTB$^+$ produces a file designated for machine reading, "results.tag", whereas when using Orca most information has to be extracted from the redirected standard output "output_orca.txt". This is done inside the function `read_output()`, which also takes the calculation mode as an argument. Then, depending on the computation, the following data is read:

- Total energy
- Gradients on the atoms
- Atomic charges
- Gradients on external charges (Orca: from file "orca.pcgrad")
- Hessian matrix (Orca: from "orca.hess")
- New atomic coordinates (in case of an optimization)*

All the information needed for the functions `e()`, `g()`, `h()`, and `o()` is then present. In chapter 4, two more virtual functions of the base class were mentioned: `charges()` and `get_g_ext_chg()`, which are necessary for integrating an energy interface into the hybrid QM/MM schemes. These functions are overridden to return the atomic charges and gradients on external charges respectively. This ensures that both interfaces are ready to be used as a part of the QM/MM approaches.

5.1.2 User-defined Options for DFTB$^+$

Above, the general workings of the interfaces to DFTB$^+$ and Orca were explained. Here, the options for DFTB$^+$, that can be defined in the CAST input file, are described in detail:

*Both programs write new coordinates into a new structure file. DFTB$^+$ uses its own so-called *gen* format to write the structure "geo_end.gen", while Orca creates a common xyz file "orca.xyz". After obtaining the atomic positions from those files, the current `Coordinates` object is modified to occupy the optimized conformation.

Option	Effect
`DFTB+path`	Path to the binary `dftb+`.
`DFTB+skfiles`	Path to the Slater-Koster files, which contain the parameters for the semi-empirical methods.*
`DFTB+verbosity`	Verbosity for DFTB$^+$. According to this value (0, 1, or 2) the program produces more or less output. If 0 is chosen, all output files are deleted after reading the information.
`DFTB+scctol`	Convergence tolerance for self consistent charge (SCC) procedure.
`DFTB+max_steps_scc`	Maximum number of steps for SCC procedure.
`DFTB+charge`	Total charge of the system.
`DFTB+3`	Activate DFTB3 (see section 3.2.4).†
`DFTB+D3`	Activate D3 dispersion correction with Becke-Johnson damping.[76]
`DFTB+D3params`	Set of damping parameters for D3 correction.‡
`DFTB+range_sep`	Activate long-range correction.[77]**
`DFTB+fermi_temp`	Activate Fermi filling with the given temperature (in K). This can improve convergence issues.
`DFTB+optimizer`	Method for local optimization. If set to 0, the L-BFGS optimizer of CAST is used. 1 and 2 correspond to the steepest descent and the conjugate gradient approach of DFTB$^+$ respectively.
`DFTB+max_steps_opt`	Maximum number of steps for optimization with a DFTB$^+$ optimizer. This option is only relevant if `DFTB+optimizer` is set to 1 or 2.

*For each element X or pair of elements X,Y present in the structure, a Slater-Koster file "X-Y.skf" is needed. They can be downloaded from the DFTB website.[38]

†For this option, a parameter set designed for DFTB3 is needed, for example 3OB. Furthermore, the hubbard derivatives for every element and the exponent ζ must be provided as additional parameters in a file called "dftb3.info", which is located in the same folder as the Slater-Koster files. Suitable values for the parameter set 3OB can also be found on the DFTB website.[38]

‡Different values for parameters a1, a2, s6, and s8 are recommended for different parameter sets. In CAST, it is possible to choose between those for 3OB, OB2(base), OB2(shift), and OB2(split). The numbers were taken from appendix F of the DFTB$^+$ manual.[64]

**This option must be combined with a parameter set specifically designed for range-separated DFTB, for example OB2.

5.1.3 User-defined Options for Orca

For the Orca-based interface, the options the user can define in the CAST input file are listed here. Further information about them can be found in the Orca manual.[65]

Option	Effect
`ORCApath`	Path to the binary `orca`.
`ORCAnproc`	Number of processors used by Orca.
`ORCAmaxcore`	Maximum amount of memory per core that is used for scratch arrays.
`ORCAmethod`	Calculation method to be applied.
`ORCAbasisset`	Basisset to be applied.
`ORCAspec`	Further specification for Orca call (this is written into the Orca input file after method and basisset verbatim).
`ORCAcharge`	Total charge of the system.
`ORCAmultiplicity`	Multiplicity of the system.
`ORCAopt`	Optimizer used for local optimizations. 0 corresponds to the L-BFGS optimizer of CAST, 1 to the internal optimizer of Orca.
`ORCAverbose`	Verbosity for Orca. According to this value (0 to 5), more or less output files are deleted after the program execution.
`ORCAcube`	Numbers of the orbitals that should be plotted as cube files.
`ORCAcasscf`	Activate CAS-SCF.
`ORCAnelec`	Number of electrons for CAS-SCF.
`ORCAnorb`	Number of orbitals for CAS-SCF.
`ORCAnroots`	Number of roots for CAS-SCF.
`ORCAnr`	Activate Newton-Raphson procedure.
`ORCAnevpt`	Activate NEVPT2.
`ORCAcpcm`	Use conductor-like polarizable continuum model (CPCM) as implicit solvent.
`ORCAeps`	Dielectric constant for CPCM.
`ORCArefrac`	Refractive index for CPCM.

This list of options is by far not exhaustive. Orca, as well as DFTB$^+$, has numerous settings that cannot be accessed via CAST at the moment. But with the general infrastructure provided, the interfaces can be quite easily expanded in the future.

5.2 Structure Input and Output Formats

For reading molecular structures, CAST uses the Tinker xyz/arc format by default.[78] Its first line contains the number of atoms, followed by an optional comment. Each of the following lines corresponds to one atom and consists of a sequential number (starting with 1), an atomic symbol (usually the element symbol), the cartesian coordinates (x, y, z) in Ångstrom, the force field atom type, and the indices of the bonding partners.[78] The following listing shows an example of a Tinker file of a water molecule:

```
3   Water
  1  O     -2.78331    1.43690    0.00000    63     2     3
  2  H     -1.81331    1.43690    0.00000    64     1
  3  H     -3.10664    0.77382   -0.62982    64     1
```

This structure format is very useful for force field calculations, as it contains the connectivity and the atom types which are needed to find the correct force field parameters in the parameter files. However, other file formats are more common in computational chemistry. In order to facilitate the use of CAST for people accustomed to them, some new input and output formats were implemented.

5.2.1 The XYZ Format

The simplest way to save a molecular structure is the xyz format.[79] Such xyz files can be created by common molecular editors like Avogadro[80] and viewed with molecular viewers like Jmol[81] or VMD[82]. This format (sometimes in a slightly modified version) is also used by quantum-chemical programs like Orca[65] or Gaussian[56]. Like the Tinker format, its title line consists of the total number of atoms. The next line can be left empty or filled with any comment. After that, there is one line for every atom, consisting of element symbol and cartesian coordinates in Ångstrom.[79] The xyz file of the water molecule above looks like this:

```
3
Water
  O          -2.78331         1.43690          0.00000
  H          -1.81331         1.43690          0.00000
  H          -3.10664         0.77382         -0.62982
```

In comparison to the Tinker format, two things are missing: The force field atom types and the bonding partners. For using a structure in CAST, atom types are only needed if a force field is used as energy interface. In all other cases, e. g. if applying one of the external energy interfaces described in section 5.1, the program also works without them. Nevertheless, an algorithm to assign atom types for the OPLSAA force field at least for some species of molecules was implemented. In order to determine which atoms are bound, a distance criterion is applied. A bond is defined between every two atoms whose distance is smaller than 1.2 times the sum of their covalent radii.[83]

Implementation of the XYZ Input Format

In CAST, every input format is implemented as a separate class. All of those classes inherit from a common base class named `input::format`, which contains only one critical virtual function to be overridden. This function is called `read()`. It takes the name of the structure file as an argument and returns a `Coordinates` object which may then be used for any kind of calculation.

A new child class of `input::format`, called `input::formats::xyz`, was implemented. Its `read()` function opens the xyz file, reads the number of atoms from the first line and discards the second line. For all the remaining lines, it saves the element symbol as a `std::string` and the coordinates as three `doubles`: `x`, `y`, and `z`. An object of type `Atom` is created from the element symbol. Its atom type is set to the dummy type value of 0 and it is added to the `atoms` vector of `coord_object`. This is the `Coordinates` object which will be returned at the end. The atomic coordinates `x`, `y`, and `z` are stored in an object of type `coords::r3` and saved into the `positions` vector of `coord_object`. After obtaining all the information directly available from the file, there is a loop over all pairs of atoms A and B, where the distance between these atoms is compared to the sum of their covalent radii. If the distance is smaller than 1.2 times this sum, A is added to the bonding partners of atom B, and B is added to the bonding partners of A. At the moment, there is only one exception to this rule: If the element symbol of one of the atoms is “Na”, this atom will form no bonds but be regarded as a single sodium ion. After creating the bonds, CAST may find atom types for the OPLSAA force field in a procedure that will be explained next.

Assigning OPLSAA Atom Types

When creating Tinker structures for force field calculations, the most time consuming part is the manual assignment of force field atom types. There are some programs that can assign atom types automatically, including Molden[84], Tinker[85] or SwissParam[86]. However, they often only accept special input formats (like mol2 in SwissParam) or use force fields that are not implemented into CAST (like the MM force fields in Molden). Some of them fail every time they come across an unrecognized structural element, which occurs quite frequently. For this reason, CAST was enhanced to be able to assign atom types, so that the resulting structure can be used in further CAST calculations. As force field, the "Optimized Potentials for Liquid Simulations (All Atom)" (OPLSAA)[54] was chosen, since it is the most versatile one of the three standard force fields implemented in CAST. The focus was on obtaining atom types for proteins. Therefore, recognition of the 20 canonical amino acids is critical. As proteins are often present in an aqueous environment, the atom types for water and simple ions like sodium should also be available. The parameter file "oplsaa.prm", from which the atom types are taken, was downloaded from the Tinker homepage.[87]

Although the atom type assignment procedure is currently only executed while reading in xyz files, it is implemented in a stand-alone class `AtomtypeFinder`. Thus, it may be included into any input format class where atom type assignment is desired. The constructor of this class takes a non-constant reference to an array of type `Atoms`, which is a vector of all atoms in the system, and saves it in a member variable. Furthermore, the member variable `got_it`, which is of type `std::vector<bool>`, is set to the same size. As this vector shows which atoms already either have an atom type or are recognized as part of an amino acid, all elements have the value `false` at the beginning. After creating an object of the `AtomtypeFinder` class, a member function `find_energy_types()` is called, which assigns atom types according to the following algorithm:

1. **Assignment of "Easy" Atom Types**
   First the atom types of simple ions and molecules receive atom types. CAST recognizes water molecules (one oxygen atom that is bound to exactly two hydrogen atoms) and sodium ions (atoms of element "Na" without any bonding partner). The corresponding fields of `got_it` are set to `true`.

2. **Finding Amino Acids**
   Amino acids are detected by their characteristic backbone consisting of a carbonyle group bound to a C_α atom with the adjacent amide nitrogen (see Figure 5.2). To find this sequence, the program first looks for an oxygen atom which is bound to one carbon atom exactly (carbonyle group). If both are not yet part of another amino acid (read from vector `got_it`), CAST looks for an adjacent carbon atom (C_α). In case this atom has a nitrogen among its bonding partners, `got_it` is set to `true` for the four backbone atoms and an object of type `AminoAcid` is created. It contains those four backbone atoms and the information whether the amino acid

is terminal. This is determined by analyzing the bonding partners of the backbone atoms: If more than one hydrogen is bound to the amide nitrogen, the amino acid is N-terminal. If there are two oxygen atoms adjacent to the carbonyle carbon, it is C-terminal.

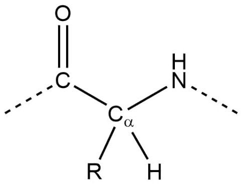


Abbildung 5.2: Backbone of an amino acid

3. **Completion of the Amino Acids**
   After the last step, each `AminoAcid` contains the four backbone atoms. Now the remaining atoms, especially those of the side chains, are added. This is done by appending all bonding partners of every atom already present in the amino acid, if they are not part of another amino acid yet. The function searching for bonds is called recursively in order to add the bonding partners of the appended atoms, too. The procedure is stopped only if there is a bond between two sulfur atoms, so cysteine residues connected by a disulfide bond remain separated. For every atom added to an amino acid, `got_it` is set to `true`.

4. **Identification of the Amino Acids**
   For each of the amino acids the chemical formula is evaluated. From this chemical formula, the amino acid is identified, and its member variable `res_name` is set to the official three-letter code or one of the additional abbreviations for protonation states introduced by Amber.[88,89] Which chemical formula corresponds to which residue name is shown in Table 5.3. If an amino acid is terminal, the chemical formula is modified: An N-terminal amino acid has one or two more hydrogen atoms, a C-terminal amino acid possesses one more oxygen and possibly also one additional hydrogen. The protonation state of the C-terminus is saved, as it will be important for the atom type assignment.
   For some amino acids, the chemical formula is ambiguous: Leucine and isoleucine are isomers and are both marked with “ILE” for the moment. Single protonated histidines can carry the proton at the N_δ or N_ϵ atom. The residue name “CYM” corresponds to deprotonated cysteines and cysteines in a disulfide bond. Amino acids not recognized obtain the residue name “XXX”.

Chemical formula	Amino Acid	Comment
C_2H_3NO	GLY	
C_3H_5NO	ALA	
$C_6H_{13}N_4O$	ARG	protonated
$C_4H_6N_2O_2$	ASN	
$C_4H_4NO_3$	ASP	deprotonated
C_3H_5NOS	CYS	
C_3H_4NOS	CYM	deprotonated or in disulfide group
$C_5H_8N_2O_2$	GLN	
$C_5H_6NO_3$	GLU	deprotonated
$C_6H_7N_3O$	HIS	single protonated
$C_6H_8N_3O$	HIP	double protonated
$C_6H_{11}NO$	ILE	Ile or Leu
$C_6H_{13}N_2O$	LYS	protonated
C_5H_9NOS	MET	
C_9H_9NO	PHE	
C_5H_7NO	PRO	
$C_3H_5NO_2$	SER	
$C_4H_7NO_2$	THR	
$C_{11}H_{10}N_2O$	TRP	
$C_9H_9NO_2$	TYR	
C_5H_9NO	VAL	

Tabelle 5.3: Amino acids and their chemical formula

5. **Atom Type Assignment for the Canonical Amino Acids**
For every amino acid that does not have the residue name "XXX", atom types are assigned. The backbone atoms are considered first, then the atoms of the side chain. The protein backbone consists of the four atoms mentioned in step 2 (carbonyle C and O, C_α and amide N), as well as any hydrogens bound to the amide nitrogen. In case of C-terminal amino acids, the second oxygen atom adjacent to the carbonyle carbon and the hydrogen atom of the carboxyle group also have to be considered. For the assignment of atom types, the identification of the specific residue is not important. Only glycine and proline have to be treated extraordinarily, as they differ in atoms directly bound to the backbone. What has to be taken into account is the terminal state of the amino acids.
On the other hand, for atoms of the side chain the terminal state is mostly irrelevant (the exception is the proline C_δ atom, which is directly bound to the amide nitrogen). For every amino acid, the atom types are determined from the element symbols and the bonding information. While analyzing the topology, the ambiguities from step 4 are resolved: Amino acids with residue name "ILE" become "LEU" if a CH_2 group

is adjacent to the C_α atom. If the sulfur atom of a “CYM” is bound to another sulfur, the residue name becomes “CYX” which means it is part of a disulfide bridge. According to their protonation state, histidines get the residue names “HID” or “HIE”.

This procedure assigns atom types to most atoms present in a common protein structure. It was tested by comparing the resulting Tinker structures to those created by the program `pdbxyz.exe` from the Tinker suite.[85] The atom types were identical for the structures tested, which included all 20 amino acids and examples for proline and glycine as termini. A big advantage of CAST over the Tinker program is that if there are unknown structure features present, `pdbxyz.exe` fails and doesn't produce any Tinker file at all, while CAST just leaves the atom types of those atoms at 0 and writes the output file. The user can thus fill the rest of the atom types manually.

Implementation of the XYZ Output Format

Besides the possibility to read in an xyz file and to convert it to a Tinker file, xyz should also be implemented as output format. Like the input formats, all output formats in CAST inherit from a common base class. It is called `output::format`. The constructor of this class takes a `Coordinates` object as an argument and saves a reference to this object in a member variable, where all the information for writing the output is taken from. Apart from the constructor, only the member function `to_stream()` needs to be provided. It is a virtual function, which receives an output stream as argument, e. g. the standard output `std::cout` or a filestream, and fills this stream with the information in the correct format.

For the xyz output, a new child class, `output::formats::xyz_cast`, was created. According to the file format described above, its overridden `to_stream()` function writes the number of atoms in the first line of the stream, a comment “Created_Using_CAST” into the second line, and the element symbol and cartesian coordinates of the atoms in the following lines. For a user, the easiest way to apply this output format is the task `WRITE_XYZ`, which just converts any input structure into an xyz file.

5.2.2 The PDB Format

Another common structure format, especially for biological macromolecules, is the Protein Data Bank (pdb) format.[90] A huge number of molecular structures in this format can be downloaded from the website of the Research Collaboratory for Structural Bioinformatics (RCSB).[91] They can be viewed with many programs like Pymol[92], Jmol[81], or

VMD[82]. The usage of CAST with the numerous structures from the Protein Data Bank was facilitated by integrating pdb as an additional input format. By including pdb as an output format as well, some of the more specific functionality of advanced viewers like Pymol was enabled with CAST output.

The pdb format contains not only the atomic structure that is needed by CAST, but also a lot of additional information, including protein sequences, secondary and tertiary structure, and metadata about the experiments from which the 3D structure was obtained. A detailed description of the file format can be found on the homepage of the world wide protein data bank (wwPDB).[90] CAST uses only the coordinate sections, whose lines characteristically start with "ATOM" or "HETATM". They contain the information about the atoms in the structure. "ATOM" describes the standard amino acids or nucleotides, "HETATM" other components like water molecules. These sections are constructed identically: Each line starts with the record name ("ATOM" or "HETATM"), followed by a sequential number for the atoms. The next field is the atom name, which might be the element symbol or a symbol that contains additional information about the topological position of the atom (like "CA" for the C_α atom). The next character indicates alternate positions the atom might occupy. Then a residue name is given, for example the name of the amino acid the atom belongs to. This field is followed by a chain identifier and a residue sequence number, i. e. a sequential enumeration of the residues. The following character is reserved for the insertion of residues. After that, there are the x-, y-, and z-coordinates of the atom, followed by the occupancy and the temperature factor obtained from the experiment. The last two fields of the line contain the element symbol and the atom charge.[90]

In many structure files, not all of this information is present, but most programs also read pdb files successfully, if some of it is missing. CAST is able to read files with only minimal data (element symbol and coordinates). For writing pdb files, CAST tries to find a balance between the data that is easily available from the `Coordinates` object and information needed to make a good use of viewers like Pymol.

Implementation of the PDB Input Format

With `input::formats::pdb`, a new child class of `input::format` was created. In this class, the function `read()` was overridden to create a `Coordinates` object from the "ATOM" and "HETATM" lines of a pdb file. As one atom may appear several times in a structure with different positions, the column with the alternate location is read and only the first appearance (location "A") is taken into account. For the creation of an atom in CAST, the element symbol is needed. If the corresponding field is not present, the program assumes that the atom name is identical to the element symbol, and builds the atom from it. This is for example the case for pdb structures saved from the software VMD. The cartesian coordinates of the atom are also read from the line. Both atoms and positions are added to the `Coordinates` object. Afterwards, bonds are added according to a distance criterion, analogously to the xyz input format (see section 5.2.1). Force field atom types are not assigned. Therefore, the pdb input can only be used with external programs as energy interface.

Structures downloaded from the Protein Data Bank usually do not contain hydrogen atoms. This is due to the X-ray diffraction method, where hydrogens can rarely be detected because of their low electron density.[93] Before using the structures in CAST, the missing atoms have to be added. This can be done for example with the command "h_add" in Pymol.[94]

Implementation of the PDB Output Format

In order to write pdb files from CAST, a new output format, `output::formats::pdb`, was implemented. It should enable some useful functionality of the Pymol viewer, especially the possibility to select and view single amino acids from the sequence and to show the secondary structure of a protein. For this, Pymol needs more information than the element symbol and position of every atom. Thus, CAST writes the following fields of the "ATOM" or "HETATM" lines: The record name, the atom serial number, the atom name, the residue name, the residue sequence number, the cartesian coordinates, and the element symbol. In order to have all this data in one place, a new `struct` called `PDBAtom` is created, which has a member variable for each of these properties. The output format class `output::formats::pdb` contains a vector of such objects, `pdb_atoms`, which is set to the size of the `Coordinates` object when the constructor is called. The vector is filled by a member function `preparation()`, which has to be called before the overridden output function `to_stream()`.

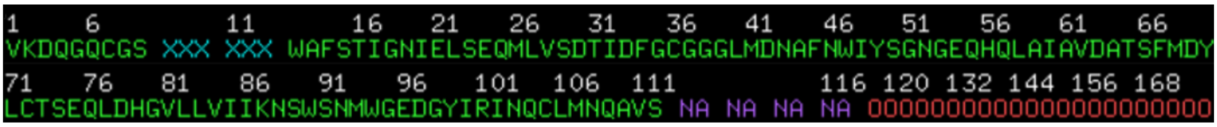
All necessary information that is not directly provided by the `Coordinates` object concerns residues. In case of proteins, they are identical to the amino acids. As finding and identifying amino acids is also part of the algorithm for assigning atom types, an object of type `AtomtypeFinder` is created. The first step in the `preparation()` function is detecting, creating and identifying `AminoAcids` according to step 2 to 4 of the algorithm described on page 74. Since Pymol needs the official three-letter codes, all of the specific Amber abbreviations have to be removed: Residues named "CYM" become "CYS", and

"HIP" becomes "HIS". Furthermore, the topology of all amino acids with residue name "ILE" has to be analyzed to determine which of them are in fact leucines. Their residue name is then changed to "LEU". With this information, the corresponding fields of the `pdb_atoms` vector can be filled for all atoms that are part of an amino acid. The record name here is "ATOM" and the atom name is the element symbol with one exception: The C_α atoms (always the 3rd atom of an amino acid, see step 2 above) are marked with "CA". This allows Pymol to recognize the protein chain, which is necessary to show the secondary structure. Finally, the elements of `pdb_atoms`, which correspond to atoms that are not part of an amino acid, have to be filled. Here, the record name is "HETATM" and molecules are counted starting from the number of amino acids to yield the residue number. The residue name is "XXX" by default, but some simple molecules and ions can be recognized analogously to the atom type assignment (step 1 in algorithm one page 74). At the moment, CAST marks water molecules with "H2O" and sodium ions with "NA".

After this preparation, the function `to_stream()` loops over `pdb_atoms` and writes every element of this vector as a line of the output stream, according to the formatting rules found on the wwPDB homepage.[90] The CAST user can create a pdb file easily by running the task `WRITE_PDB`.

Example: Complex of Rhodesain and K11777

As an example, the enzyme inhibitor complex between rhodesain and K11777 is used. The formation and dissolution of this complex will be the topic of chapter 6. An xyz file containing a section of it was converted to pdb format via CAST and the result is shown as visualized in Pymol.



```
1    6       11      16   21   26   31   36   41   46   51   56   61   66
VKDQGQCGS XXX XXX WAFSTIGNIELSEQMLVSDTIDFGCGGGLMDNAFNWIYSGNGEQHQLAIAVDATSFMDY
71   76   81   86   91   96   101  106  111         116 120 132 144 156 168
LCTSEQLDHGVLLVIIKNSWSNMWGEDGYIRINQCLMNQAVS NA NA NA NA OOOOOOOOOOOOOOOOOOOOOO
```

Abbildung 5.3: Sequence of the enzyme inhibitor complex as shown in Pymol

The sequence as shown in Pymol is depicted in Figure 5.3. It consists of 112 amino acids, followed by 4 sodium ions and many water molecules (sequence is cut). For most of the amino acids, the one-letter codes are shown. Only residues 10 and 11 are not recognized and marked with their full residue name "XXX" in the sequence. Upon closer inspection, they correspond to the cysteine which binds the inhibitor and the inhibitor itself. As the cysteine side chain is modified, CAST is not able to identify the amino acid as cysteine by the chemical formula. The inhibitor is erroneously assumed to be an amino acid because it contains the characteristic backbone. In spite of these flaws, it is possible to create suitable illustrations from the resulting pdb file as shown in Figure 5.4 .

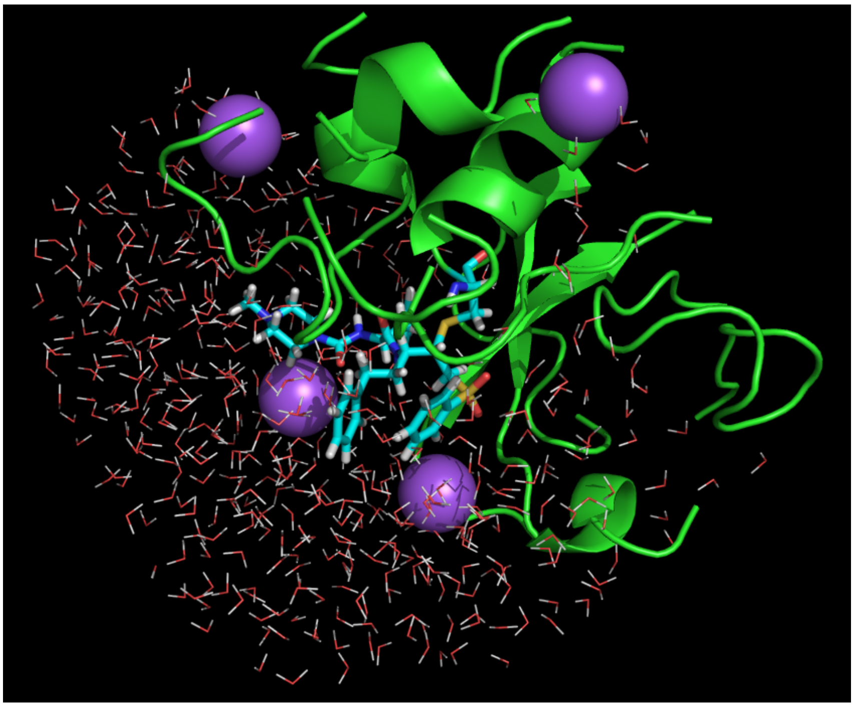


Abbildung 5.4: Illustration of the enzyme inhibitor complex in Pymol

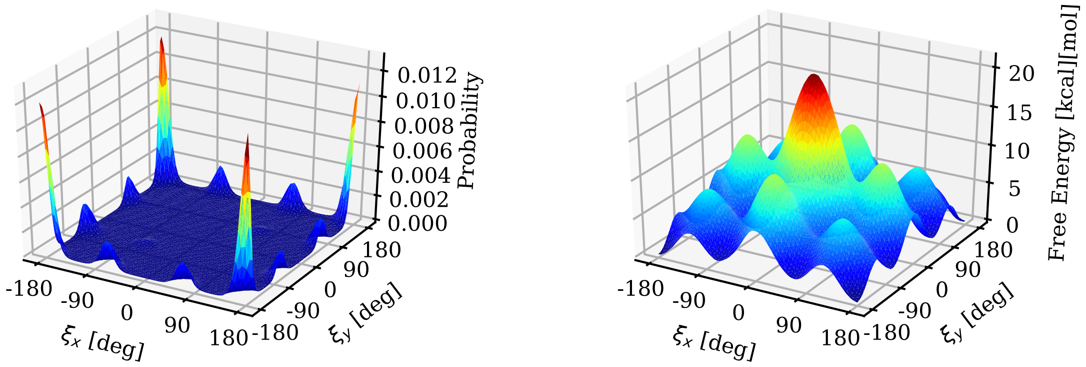
5.3 Extensions of the Umbrella Sampling Task

The original implementation of the Umbrella Sampling task was developed by Johannes Becker as part of his PhD thesis.[48] As described in section 3.3, Umbrella Sampling consists of two steps: First, MD simulations are performed at different positions on the reaction path, retained by an additional bias potential. Then, the results of these simulations are combined using the WHAM technique to obtain the unbiased probability distribution and thus the relative free energies along the path. In CAST, only the first step is implemented, i. e. a simulation with an additional harmonic potential on the reaction coordinate is run. The reaction coordinate ξ can be defined as any atomic distance, angle, or dihedral angle. The output of this task is a file called “umbrella.txt”, which contains the value for ξ at every production step of the simulation (after equilibration). These output files, one for each window, serve as input files for the external program WHAM[95], which performs the analysis of the data according to the weighted histogram method.

5.3.1 Two-dimensional Umbrella Sampling

Umbrella sampling as described in section 3.3 produces a free energy path along a reaction coordinate ξ. It is also possible to apply it on two different reaction coordinates, ξ_x and ξ_y, to create a two-dimensional free energy surface.[96] In the first step, two biases are added to the total potential function and the values of both reaction coordinates at every step are saved into the output file. These files then serve as input for the binary `wham-2d`, which is also part of the WHAM program suite and performs a two-dimensional weighted histogram analysis.[96]

As an example for two-dimensional Umbrella Sampling, the rotation of the backbone dihedrals of pentane was chosen. This is the logical expansion of the butane example in section 3.3.4. The calculation was performed with analogous options. Therefore, both reaction coordinates, corresponding to the two backbone dihedrals of pentane, were partitioned into 73 windows between $-180°$ and $180°$. Since all combinations of ξ_x and ξ_y have to be sampled, this means that in total 5329 MD simulations need to be performed. The force constant for the bias is $0.05\frac{\text{kcal}}{\text{mol}\cdot\text{deg}^2}$. Every simulation consists of 10000 equilibration and 500000 production steps. The CHARMM force field was used as energy interface. The temperature was kept constant at 300 K with the Nosé-Hoover thermostat.[97]



(a) Probability

(b) Relative Free Energy

Abbildung 5.5: Results for the Umbrella Sampling of the pentane rotation

The unbiased probability distribution and the free energy surface computed by WHAM are depicted in Figure 5.5. They look as expected from comparison with the butane example: The highest probability (and lowest energy) is found when both dihedrals are in the anti-position, i. e. the angle is ±180°. The highest free energy is at the center of the plot, where both reaction coordinates have a value of 0°. This point is surrounded by 8 minor spikes that resemble the partially ecliptic conformations at ±60°.

When performing such a two-dimensional sampling, one is often interested in the change of free energy along a certain pathway. In this case, it is a waste of computational resources to sample the whole surface. Instead, it should be possible to perform only those simulations that are necessary to cover the path. To test this, the PMF for a path where both backbone dihedrals of pentane go simultaneously from −180° to 180° was evaluated. This corresponds to a diagonal way through the surface. The result is shown in Figure 5.6. The black line was drawn using the free energy values from the WHAM analysis of all simulations (Figure 5.5). A second WHAM analysis was performed using only the data from the windows directly on the path ($\xi_x = \xi_y$) in order to get the blue curve. Though the number of necessary MD simulations was reduced from 5329 to 73, there is hardly any change in the potential of mean force.

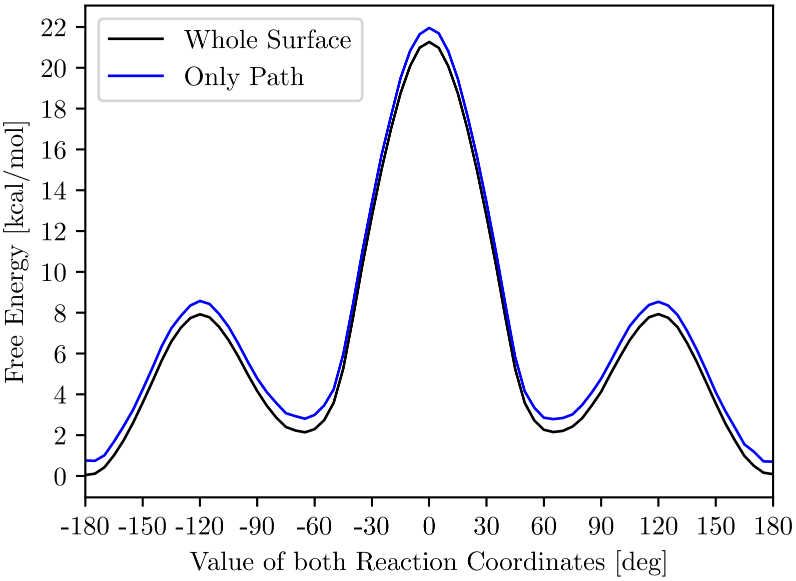


Abbildung 5.6: Comparison of reaction paths calculated by Umbrella Sampling taking into account all windows or only those where $\xi_x = \xi_y$

5.3.2 Linear Combinations of Distances as Reaction Coordinate

Some reactions cannot be characterized by just one atomic distance or (dihedral) angle, but require more complex movements to describe the reaction path.[98] One possibility to create better reaction coordinates is using a linear combination of atomic distances.[98] A quite simple example is a transfer reaction, as shown in Figure 5.7. If the reaction coordinate is defined as $\xi = d_{AB} - d_{BC}$, the reaction can be described by a growing ξ. However, the system may either increase the distance between A and B first (dissociative mechanism), or decrease the distance between B and C (associative mechanism). So by running an Umbrella Sampling with such a combined reaction coordinate, it is possible to discriminate between different reaction mechanisms.[98]

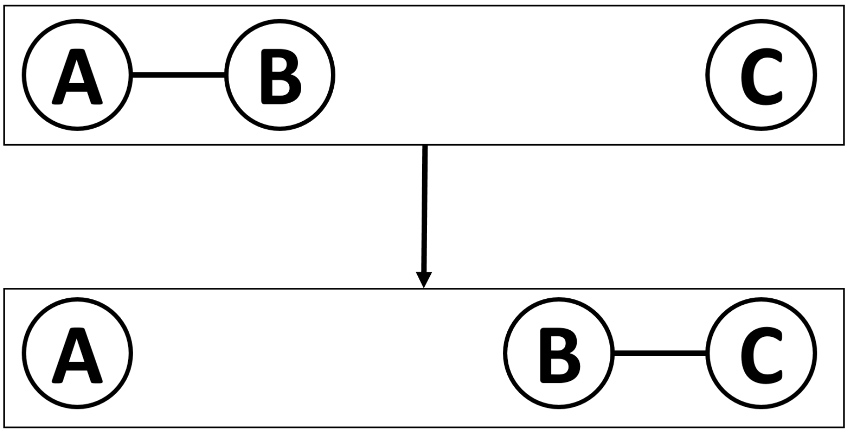


Abbildung 5.7: Illustration of a transfer reaction

To use this kind of reaction coordinate in CAST, a possibility to apply a bias potential according to Equation 3.86 on it was implemented. With ξ defined as a linear combination of distances

$$\xi = \sum_j f_j \cdot d_j \tag{5.1}$$

where f_j is an arbitrarily chosen prefactor (in the above example it is 1 for d_{AB} and -1 for d_{BC}), the bias becomes:

$$\omega_i(\xi) = \frac{1}{2}K\left(\sum_j f_j \cdot d_j - \xi_i\right) \tag{5.2}$$

Since every d_j is defined as the distance between two atoms $J1$ and $J2$

$$d_j = \sqrt{|R_{J1} - R_{J2}|^2} \tag{5.3}$$

the additional gradients caused by the potential ω_i can be calculated by differentiating

Equation 5.2 with respect to all atomic coordinates R_{J1} and R_{J2}:

$$\frac{\partial \omega_i}{\partial R_{J1}} = K(\xi - \xi_i) f_j \cdot \frac{1}{d_j}(R_{J1} - R_{J2}) \tag{5.4}$$

$$\frac{\partial \omega_i}{\partial R_{J2}} = -K(\xi - \xi_i) f_j \cdot \frac{1}{d_j}(R_{J1} - R_{J2}) \tag{5.5}$$

The option to apply a linear combination of atomic distances as reaction coordinate for the Umbrella Sampling task was added to CAST. The user can choose any number of distances that should be included, as well as the prefactor f_j for each of these distances. For stability reasons, the force constant for the bias is set to zero at the beginning and grows linearly during the first half of the equilibration steps until it reaches the desired value. Then it is kept at this value for the second half of the equilibration and the entire production run.

This new option was tested on the exchange of the chlorine atom in chloromethane, which is a well-known example in literature:[99]

$$\mathrm{Cl^- + CH_3Cl \rightarrow CH_3Cl + Cl^-} \tag{5.6}$$

As the PMF proved to be very sensitive to the choice of the starting structures, three different structures were created that resembled the starting point, the transition state, and the endpoint of the reaction. For the starting point structure (*original*), the system consisting of chloromethane and chloride was drawn and optimized in Avogadro.[80] Then, the `SOLVEADD` task in CAST was used to add 796 water molecules without any further optimization.[100] By manually exchanging the two chlorine atoms in the resulting file, the structure corresponding to the endpoint (*reverse*) was obtained. The transition state (*zero*) was created in GaussView, where it is possible to define the internal coordinates (i. e. atom distances and angles) in a z-matrix format.[101] According to literature, the C-Cl distances were both set to 2.4 Å.[99] Then, 796 water atoms were again added using `SOLVEADD`.

The reaction coordinate ξ for the Umbrella Sampling was defined as the difference between the distances of the central carbon atom to the two chlorine atoms. 39 windows were sampled, where ξ was equally distributed between -3.8 Å and 3.8 Å. The force constant for the bias was $50\frac{\mathrm{kcal}}{\mathrm{mol}\cdot\text{Å}^2}$. The *original* starting structure was applied to the windows with $\xi < -0.4$ Å, the *reverse* one to those with $\xi > 0.4$ Å. The five simulations in between started from structure *zero*.

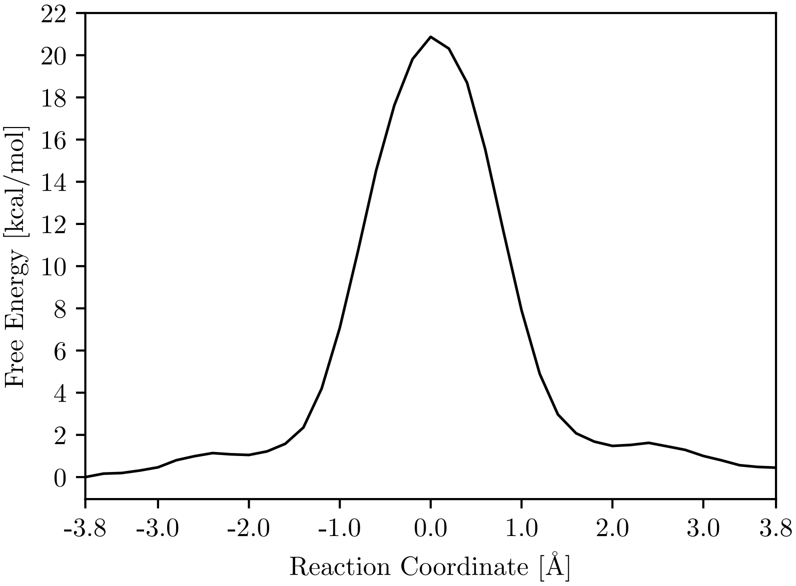


Abbildung 5.8: PMF for the chloromethane reaction

Every MD consists of 20000 equilibration steps and another 20000 production steps. The additive QM/MM scheme was chosen as energy interface. The QM region, consisting of the chloromethane molecule and the chloride ion, is computed with DFTB3. The surrounding water molecules are treated with the OPLSAA force field. During the simulations, all atoms outside of a 13 Å sphere around the central carbon were fixed. The temperature was kept constant at 300 K using a Nosé-Hoover thermostat.

The resulting PMF is shown in Figure 5.8. It agrees well with those from literature that were obtained with similar parameters.[99] At very high and low values of ξ, the relative free energy is around 0. It starts rising at about $\xi = \pm 2\,\text{Å}$, up to a maximum of $20.9\frac{\text{kcal}}{\text{mol}}$ at the transition state ($\xi = 0$). The barrier is lower than in the cited literature, where it is approximate $26\frac{\text{kcal}}{\text{mol}}$. However, there are some changes in the calculation parameters, which may account for the observed difference. The most important of them is the method for the computation of the active site (DFTB3 instead of HF/6-31G*).

To investigate the reaction mechanism, the average distance of the carbon atom to the two chlorine atoms in every simulation was plotted (Figure 5.9). The curves indicate an associative mechanism, as the distance to the incoming chlorine ion decreases (d_2, red line) before the other chlorine atom leaves the molecule (d_1, blue line). Such an S_N2 reaction was also found by previous research.[102] At the transition state, both distances are around 2.4 Å, which is also in perfect agreement with the results reported in literature.[99]

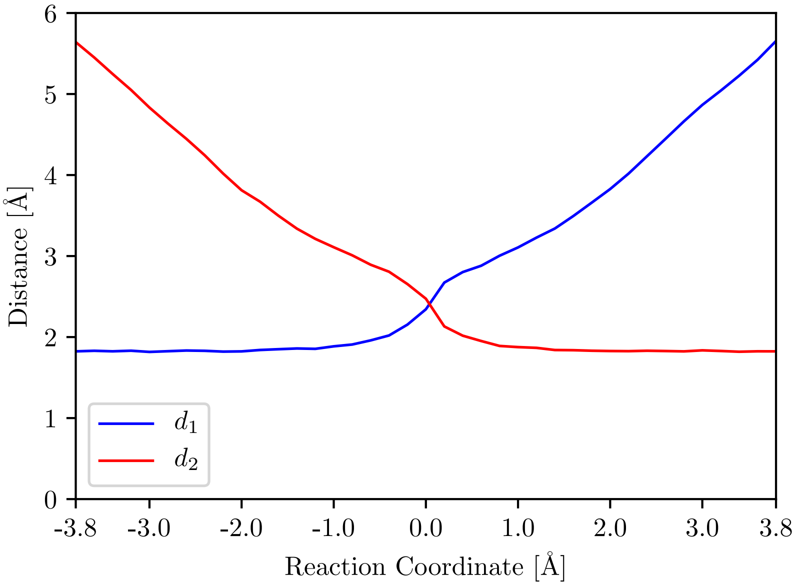


Abbildung 5.9: Average distances of the carbon atom to the chlorine atoms during the MD simulations

5.3.3 Potentials of Mean Force with Interpolated Corrections

In order to enhance the results of an Umbrella Sampling by a better energy evaluation method, the interpolation correction as described in section 3.3.5 was implemented into CAST. Three steps are necessary for applying this correction:

1. Calculation of singlepoint energies on a number of structures along the reaction path, once with the low-level energy method used for the Umbrella Sampling and once with the high-level correction method.

2. Creation of a spline function from the energy differences.

3. Performing the MDs for the Umbrella Sampling with this spline function as an additional potential.

The first step can be done manually by running single computations on the structures and collecting the energy values. To facilitate this, a task `PMF_IC_PREP` was added to CAST. It takes a variable number of structures and evaluates the energy with the high-level and the low-level method as well as the difference ΔE (see Equation 3.99). Furthermore, the user has to define the reaction coordinate ξ by giving the atom indices of the partaking atoms. The character of the reaction coordinate (distance, angle or dihedral) is determined by the number of atoms involved. CAST calculates the value of ξ in every structure from the atomic coordinates and saves it along with E_{HL}, E_{LL} and ΔE into the output file.

In order to enable the user to check if the spline looks reasonable, this task also provides the possibility to get a preview of it. The mapping variable z is computed for every structure according to Equation 3.100 with the user-defined parameters ξ_0 and L. The spline is created by the C++ library Alglib[103]. Its function `spline1dbuildcubic()` builds a cubic spline $S(z)$ from the values for z and ΔE. The result of the spline function for any z can be obtained now by using the function `spline1dcalc()`, which is also provided by Alglib. For a range of ξ values given by the user, the program converts them to z and writes the value for $S(z)$ into a csv file. This file can be viewed in any spreadsheet software (e. g. LibreOffice Calc[104]) to check whether the spline function is reasonable for the function at hand.

The second and third step of the procedure outlined above are performed during the Umbrella Sampling task. If the interpolated correction is activated, CAST requires a file containing values for ξ and ΔE that is formatted like the one produced by the `PMF_IC_PREP` task. During the initialization of the umbrella simulation, this file is read and a spline is created as described above. The spline is added to the potential function during the MD simulation. According to Equation 3.102, this causes an additional gradient of:

$$\frac{\partial S}{\partial R} = \frac{\partial S(z)}{\partial z} \cdot \frac{\partial z}{\partial \xi} \cdot \frac{\partial \xi}{\partial R} \tag{5.7}$$

The first part, the derivative of the spline with respect to z, can be obtained from the Alglib function `spline1ddiff()`. To get the second part, Equation 3.100 has to be differentiated:

$$\frac{\partial z}{\partial \xi} = \frac{2L}{\pi(\xi_0^2 - 2\xi_0\xi + L^2 + \xi^2)} \tag{5.8}$$

The third part depends on the character of the reaction coordinate ξ. If ξ corresponds to an atomic distance $d = R_A - R_B$, the derivative with respect to the coordinates of the atoms A and B are:

$$\frac{\partial d}{\partial R_A} = (R_A - R_B) \cdot d^2 \tag{5.9}$$

$$\frac{\partial d}{\partial R_B} = -(R_A - R_B) \cdot d^2 \tag{5.10}$$

If ξ is an angle $\alpha = \frac{P}{d_{AB} \cdot d_{BC}}$ with the scalar product $P = (R_A - R_B) \bullet (R_C - R_B)$, the derivative $\frac{\partial \xi}{\partial R}$ evaluates to:

$$\frac{\partial \alpha}{\partial R_A} = \frac{-1}{\sqrt{1-\cos^2\alpha}} \cdot \left(\frac{P \cdot (R_B - R_A)}{d_{AB}^3 \cdot d_{BC}} + \frac{R_C - R_B}{d_{AB} \cdot d_{BC}} \right) \tag{5.11}$$

$$\frac{\partial \alpha}{\partial R_B} = \frac{-1}{\sqrt{1-\cos^2\alpha}} \cdot \left(\frac{P \cdot (R_A - R_B)}{d_{AB}^3 \cdot d_{BC}} + \frac{P \cdot (R_C - R_B)}{d_{AB} \cdot d_{BC}^3} + \frac{2R_B - R_A - R_C}{d_{AB} \cdot d_{BC}} \right) \tag{5.12}$$

$$\frac{\partial \alpha}{\partial R_C} = \frac{-1}{\sqrt{1-\cos^2\alpha}} \cdot \left(\frac{R_A - R_B}{d_{AB} \cdot d_{BC}} + \frac{P \cdot (R_B - R_C)}{d_{AB} \cdot d_{BC}^3} \right) \tag{5.13}$$

In case of a torsional angle φ between the four atoms A, B, C, and D, the derivatives are written with the help of intermediate vectors defined as $F = R_A - R_B$, $G = R_B - R_C$, $H = R_D - R_C$, $I = F \times G$ and $J = H \times G$:[105]

$$\frac{\partial \varphi}{\partial R_A} = -\frac{|G|}{I^2} I \tag{5.14}$$

$$\frac{\partial \varphi}{\partial R_B} = \frac{|G|}{I^2} I + \frac{F \bullet G}{I^2 |G|} I - \frac{H \bullet G}{J^2 |G|} J \tag{5.15}$$

$$\frac{\partial \varphi}{\partial R_C} = \frac{H \bullet G}{J^2 |G|} J - \frac{F \bullet G}{I^2 |G|} I - \frac{|G|}{J^2} J \tag{5.16}$$

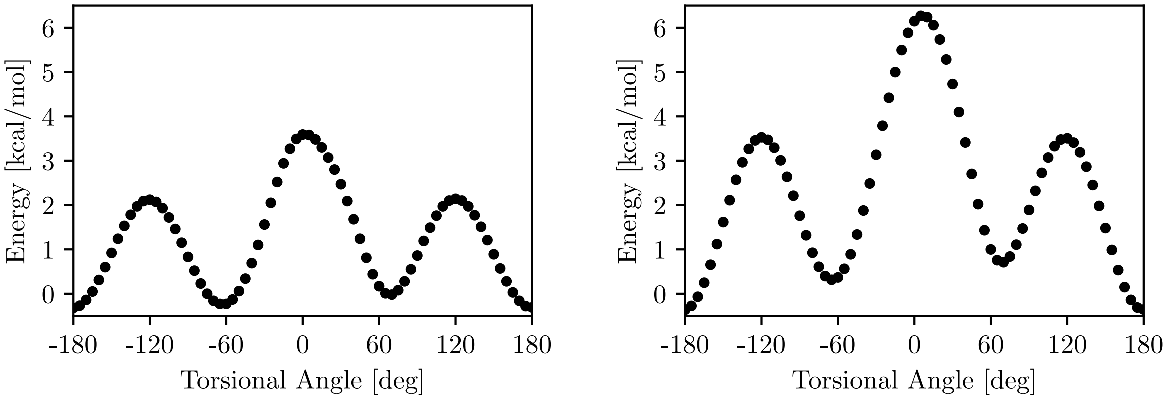
$$\frac{\partial \varphi}{\partial R_D} = \frac{|G|}{J^2} J \tag{5.17}$$

Using Equations 5.7 to 5.17, the additional gradients on the atoms involved in the reaction coordinate can be calculated. They are applied in all MD simulations performed during Umbrella Sampling in order to get a Potential of Mean Force with Interpolation Correction (PMF-IC).

Example: Rotation of Butane

In order to test the implementation of PMF-IC, the PMF for the rotation of butane (see section 3.3.4) will be corrected using singlepoint calculations at the DFTB3 level of theory. For this purpose, 73 conformations of butane were created, where the backbone dihedral was equally distributed between −180° and 180°. The energy of these conformers was evaluated by the task `PMF_IC_PREP` with both the high-level (DFTB3) and the low-level method (CHARMM). The energy difference $\Delta E = E_{HL} - E_{LL}$ was obtained. The relative singlepoint energies are depicted in Figure 5.10. The general shape remains the same for both methods, but the height of the rotational barrier is predicted much lower invoking DFTB3. For this reason, the correction energy at the ecliptic conformations (0°, ±120°) is more negative than at the gauche or anti-conformations (see Figure 5.11a).

To create the spline, the mapping parameters were set to $\xi_0 = 0°$ and $L = 90°$. The

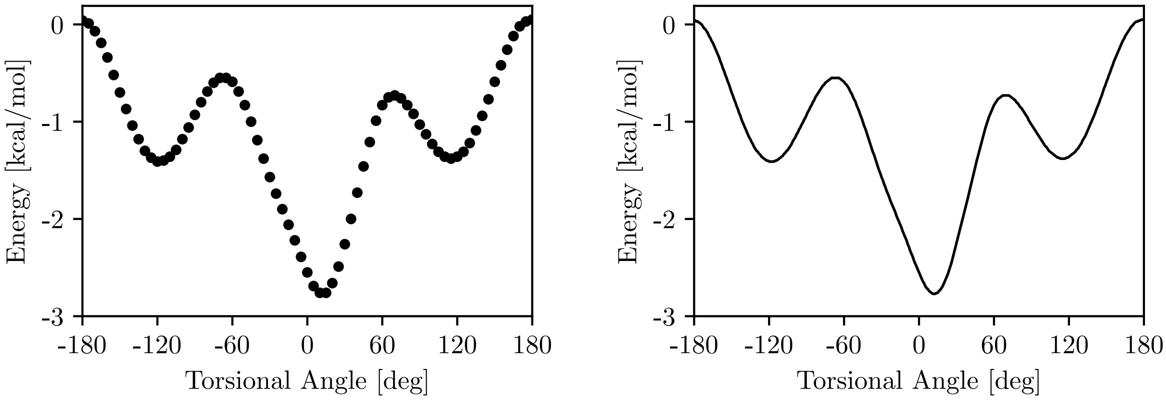


(a) DFTB3 (high level)

(b) CHARMM (low level)

Abbildung 5.10: Singlepoint energies of butane during rotation

spline function is plotted in Figure 5.11b. In comparison to the graph shown in Figure 5.11a, it is clear that the spline models the given points for the energy correction well.



(a) $\Delta E = E_{HL} - E_{LL}$

(b) Spline Function

Abbildung 5.11: Energy difference ΔE in comparison to spline function

As the last step, the Umbrella Sampling is started again with this spline added to the CHARMM energy and gradient function. The rest of the options remain the same as in section 3.3.4: The reaction is divided into 73 windows, where the molecule is restrained by a bias potential with a force constant of $0.05\frac{\text{kcal}}{\text{mol}\cdot\text{Å}^2}$. Each of the simulations consists of 10000 equilibration and 500000 production steps and is run at 300 K. The resulting PMF is shown in Figure 5.12 (blue line). The energy barrier is much lower than in the computation without interpolation correction (black line). Regarding the singlepoint energies in Figure

5.10, this is the expected behavior.

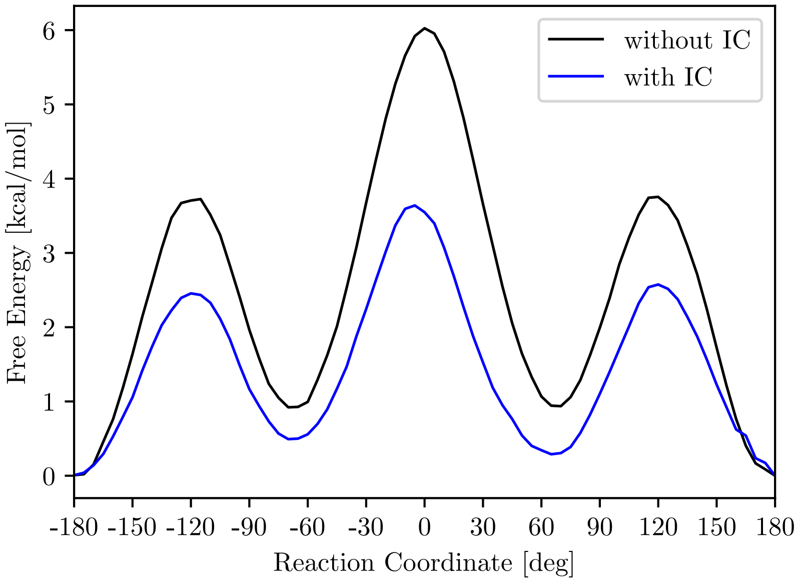


Abbildung 5.12: PMF for the rotation of butane with and without interpolation correction

Extension to Two Dimensions

In order to apply the interpolation correction also on two-dimensional Umbrella Sampling, a two-dimensional spline function is needed. It depends on both reaction coordinates ξ_x and ξ_y or – more specifically - on the mapping variables z_x and z_y obtained by Equation 3.100. Such bicubic splines are also provided by Alglib.[103] When constructing bicubic splines from datapoints, there are two options:[106] If the points form a grid on the xy plane, the spline can be created by interpolation. Then the spline is composed of rectangular patches between the gridpoints. If, however, the data is irregularly distributed over the xy plane, the spline has to be created by least square fitting. Since the spline function is still constructed of rectangular patches, the grid has to be defined by the user. If the spline function has strong fluctuations, it is possible to give a further parameter, the so-called *nonlinearity penalty*, to render the function more rigid.[106]

Preferably, CAST is able to calculate the correction function from arbitrary structures, without needing them to form a grid. Thus, the second option, spline fitting, is used. Some additional parameters have to be given by the user in order to create the spline: The number of gridpoints in x- and y-direction, the nonlinearity penalty, and the mapping parameters ξ_0 and L for both dimensions. The task `PMF_IC_PREP` computes the value for the reaction coordinates ξ_x and ξ_y as well as E_{HL}, E_{LL} and the difference ΔE for every given structure. Furthermore, it constructs the spline from the above parameters with the BlockLLS solver from Alglib[106] and writes it into a csv file for preview. During the Umbrella Sampling task, this spline is added to the energy. The additional gradients caused by it are computed analogously to Equation 5.7 by differentiation with respect to the two reaction coordinates:

$$\frac{\partial S}{\partial R} = \frac{\partial S(z_x, z_y)}{\partial z_x} \cdot \frac{\partial z_x}{\partial \xi_x} \cdot \frac{\partial \xi_x}{\partial R} + \frac{\partial S(z_x, z_y)}{\partial z_y} \cdot \frac{\partial z_y}{\partial \xi_y} \cdot \frac{\partial \xi_y}{\partial R} \tag{5.18}$$

The spline derivatives $\frac{\partial S(z_x,z_y)}{\partial z_x}$ and $\frac{\partial S(z_x,z_y)}{\partial z_y}$ are provided by the Alglib library through the function `spline2ddiff()`. The derivative of z_x and z_y is calculated by Equation 5.8, and derivatives of ξ_x and ξ_y according to the character of the reaction coordinate are obtained by Equations 5.9 to 5.17.

Example: Rotation of Hexane

The two-dimensional interpolation correction is exemplified using the hexane molecule. The reaction coordinates are defined as the backbone dihedrals φ_{1234} and φ_{3456}, i. e. those containing end-group carbons. The spline is created from singlepoint calculations of 5327 structures, where both torsions vary between $-180°$ and $180°$ with a steplength of 5°. To create the mapping variables, ξ_0 was set to 0° and L to 90° for both dimensions. The grid consisted of 73 points in each direction and the nonlinearity penalty was 0.1. The spline corresponding to $\Delta E = E_{DFTB3} - E_{CHARMM}$ is depicted in Figure 5.13.

Then, Umbrella Sampling was performed once with and once without applying this spline as correction function. The parameters are identical to those for the pentane example in section 5.3.1. The resulting free energy surfaces are shown in Figure 5.14. The height of the peak at $\xi_x = \xi_y = 0°$ is much lower with the correction by DFTB3 ($5.4\frac{\text{kcal}}{\text{mol}}$ vs $12.9\frac{\text{kcal}}{\text{mol}}$), and the difference to the minor spikes at torsional angles of ±120° decreases. This is similar to the one-dimensional butane example.

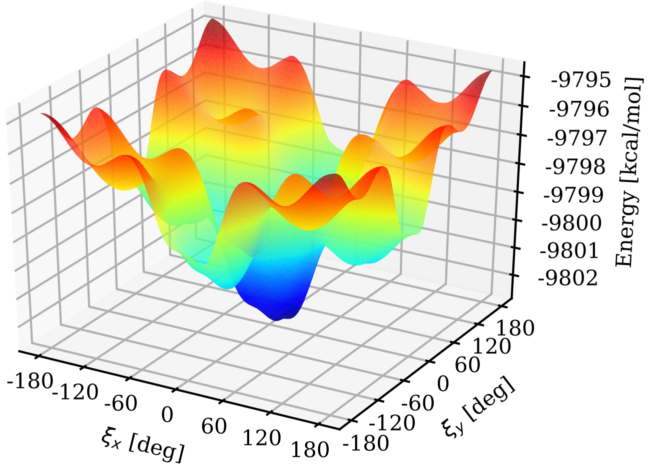
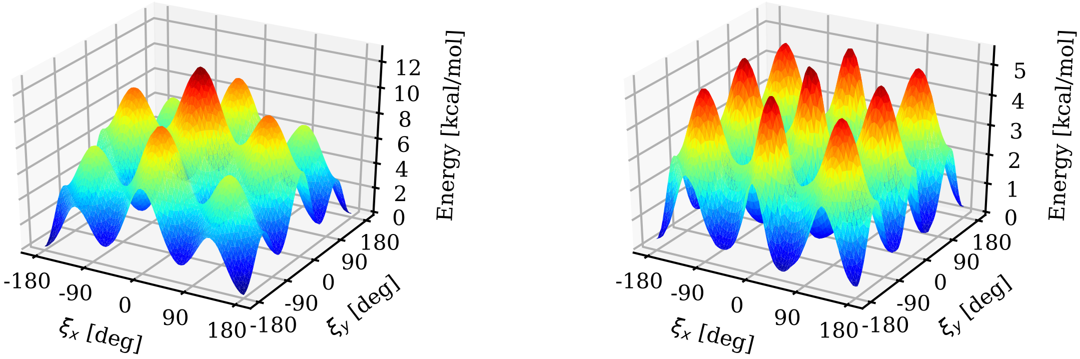


Abbildung 5.13: Spline function for the rotation of hexane



(a) without IC

(b) with IC

Abbildung 5.14: PMF for the rotation of hexane

5.4 Optimization Methods

When a local optimization in CAST is performed, the member function `o()` of the `Coordinates` class is called. If the energy interface contains its own optimizer routine, this routine is subsequently executed. This is the case for most external programs, like Orca or DFTB$^+$ (see section 5.1). If the interface does not have its own optimization method, e. g. in case of the force fields, one of the general optimizers implemented in CAST is applied. As they only require energies and gradients to perform a minimization, they can be used in combination with any energy interface. The default optimizer in CAST is the Limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS).[107] In recent versions of CAST, there is also the option to perform the optimization in Translation Rotation Internal Coordinates (TRIC), which can improve convergence in comparison to cartesian coordinates.[83]

5.4.1 Microiterations in QM/MM

During a local optimization with the default L-BFGS algorithm, the energy and gradients of the system have to be evaluated in every minimization step. If one of the QM/MM schemes is chosen as an energy interface, the system consists of a large MM region that is inexpensive to calculate, and a small QM region whose computation is the most time-consuming step.[108] For this reason, it might be more efficient to optimize the geometry of the two regions independently of each other in order to minimize the number of QM calculations, even if this causes a higher number of MM evaluations. A practical solution is the so-called microiteration scheme, which is implemented in several third-party QM/MM software packages.[108]

In the subtractive QM/MM scheme, the energy (and gradients) are obtained from three independent computations (see Equation 4.15). The second and third term of this equation only include the *small* system, consisting of QM and link atoms. The MM atoms are only present in the first term, $E_{big,MM}$. The only exception are the MM atoms directly bound to the QM region (M1 atoms), as they influence the *small* system through the position of the link atoms (Equation 4.6).[108] So the first step of the optimization procedure is to minimize the energy of the total system with the MM interface, while all QM atoms and the M1 atoms are fixed. This is called the **microiterations**. Then, the atoms moved in the first step are fixed and the rest of the atoms (i. e. QM and M1 atoms) is optimized with the energy and gradients from the QM/MM scheme during the so-called **macroiterations**.[109] As $E_{big,MM}$ (and the corresponding gradients) depends on the positions of the QM atoms, the forces on the MM atoms may not be converged anymore. Therefore, a new cycle of micro- and macroiterations starts. This is repeated until overall convergence is reached.[108]

For mechanical embedding, the assumption that the calculation of the *small* system does not influence the forces on the MM atoms is exact.[108] However, if electrostatic embedding is applied, this is not the case anymore, because the MM atoms are included into the *small* computations as external charges. Since the QM atoms cause forces on these external charges, the PES during the micro- and the macroiterations is not the same. This hampers convergence. Several approaches have been proposed to address this issue by modification of the force field gradients to fit them to the QM/MM surface:[108] For the calculation of the electrostatic interactions, the charges for the QM atoms might be taken from the QM computation instead from the charge parameters:[110]

$$G_{coul,MM} = \sum_{interactions} \frac{\partial}{\partial R_{MM}} \frac{Q_{MM} \cdot Q_{QM}}{d_{QM-MM}} \tag{5.19}$$

This works quite well if the QM computation is performed using a semi-empirical program that also evaluates the interaction with the external charges as Coulomb interactions between those charges and the atomic charges. But most QM software includes the external charges by evaluating their interactions with nuclei and electrons of the atoms. Thus, the surfaces are still different.[108] Another possibility is to apply the exact Coulomb gradients from the QM/MM calculation at geometry R_0 before the relaxation of the MM atoms, and describe only the perturbation caused by the movement of the MM atoms by the interactions with the QM charges:[111]

$$G_{coul,MM} = G_{coul,QM/MM}(R_0) + \sum_{interactions} \left(\frac{\partial}{\partial R_{MM}} \frac{Q_{MM} \cdot Q_{QM}}{d_{QM-MM}} - \frac{\partial}{\partial R_{MM}} \frac{Q_{MM} \cdot Q_{QM}}{d_{QM-MM}(R_0)} \right) \tag{5.20}$$

When approximating convergence, the magnitude of the correction term is decreasing and the PES at the point of the optimized structure is identical for both MM and QM/MM calculations.[111] At the beginning of the optimization, however, this correction term is growing during the microiterations. Thus, the MM atoms are not fully optimized with respect to the QM/MM surface at the outset of the macroiterations.[108] Therefore, this scheme can be improved by re-evaluating new starting gradients $G_{coul,QM/MM}(R_0)$ from a singlepoint QM/MM calculation on the structure obtained after the microiterations and performing a new MM relaxation using them. This is repeated until self-consistency is reached. Only then, the next round of macroiterations is performed.[108]

Implementation

As this microiteration scheme is a specific optimization algorithm for QM/MM methods, it was implemented in the `o()` function of the subtractive energy interface in CAST. The subtractive interface was chosen because it already contains `mmc_big`, a `Coordinates` object of all atoms including a pointer to the MM interface. This is necessary for the execution of microiterations. The first step of the optimization is the fixation of all QM atoms and those directly bound to them (M1), and a minimization of `mmc_big` using the L-BFGS optimizer. Then the geometry of `mmc_big` is transferred to the overall `Coordinates` object, which holds a pointer to the `QMMM_S` interface. After fixing the MM atoms (except M1), the macroiterations are run, i. e. an optimization of this object with the L-BFGS algorithm is performed. If convergence is not reached yet, the optimized geometry is transferred to `mmc_big` and another round of microiterations starts. Otherwise, the energy of the optimized structure is returned.

The convergence criterion is determined from the gradients after each iteration cycle. If the maximum component of the gradients is smaller than a user-given value c, and the root-mean-square (RMS) of the overall gradients vector is smaller than $\frac{2}{3}c$, the system is considered converged, and the optimization stops.[109] Convergence is also assumed if both the number of micro- and macroiterations in the last cycle is zero, i. e. no optimization takes place anymore.

To improve the behavior of the optimization with electrostatic embedding, the modification of the MM interface according to Equation 5.19 was implemented, too. If this option is chosen, the energy interface of `mmc_big` does not use the default force field parameters for calculating the Coulomb interactions. Instead, single atom charges are applied, which must be set previously. Before the microiterations this option has to be switched on, and after performing them it is switched off for the macroiterations. While it is activated, the charges are taken from the vector `atom_charges`, a member variable of the current `Coordinates` object `mmc_big`. It is filled through the use of the `charges()` function of the `QMMM_S` interface, which is overridden to return the QM charges for all atoms that are part of a QM system, and the charge parameters for the rest of the atoms. All charges are obtained from the `charges()` function of the respective QM or MM interface. The return value can be directly taken as `atom_charges` for the evaluation of the modified Coulomb interactions in the force field interface.

An analogous optimization scheme was also implemented for the three-layer interface (see section 4.2.1). The system `mmc_big` is used for the microiterations, where all atoms of the *medium* system, as well as those directly bound to it, are fixed. The macroiterations are applied to the overall `Coordinates` object that contains a pointer to the three-layer interface, while only the previously fixed atoms are allowed to move.

The adjustment of the charges for the MM interface during the microiterations is done analogously to the two-layer scheme. The `charges()` function of the three-layer interface is overridden to return QM charges for the QM atoms (from `qmc_small`), SE charges for the SE atoms (from `sec_medium`), and MM charges for the MM atoms (from `mmc_big`). If the return value of this function is used as `atom_charges` in the force field, this corresponds to a description of the interactions between SE and MM atoms on SE level and between QM and MM atoms on QM level. This holds true for electrostatic embedding with options EE, EE+ and EE+X, where the MM atoms are included in *medium* and *small* system as external charges, but not for EEx, where no external charges are added to the *small* system. In this case, the interactions between QM and MM atoms are treated on the SE level, since the MM charges are included in the *medium* system which consists of QM and SE atoms. For this reason, the atom charges used for the microiterations may not be directly taken from the `charges()` function if EEx is chosen. Instead, the charges for the QM atoms have to be replaced by those taken from `sec_medium` before running the minimization.

Example: Complex of Rhodesain and K11777

The microiteration method was tested on a complex of rhodesain and inhibitor K11777 (see chapter 6). The structure was taken from entry 2p7u of the Protein Data Bank.[91] Conversion to the Tinker format and assignment of OPLSAA atom types were done using the tools described in section 5.2. The QM region consists of a part of the inhibitor and the sidechains of the amino acids Cys25 and His162. These 43 atoms are described by the QM method DFTB3, while the OPLSAA force field was used for the remaining 3630 atoms. The subtractive QM/MM scheme with electrostatic embedding was applied as energy interface. Only those MM atoms directly bound to the QM system were not taken into account when creating the external charges (*delM1*).

The energy of this system was minimized by the previously outlined optimization scheme with and without an adjustment of the charge parameters for the microiterations. For comparison, this minimization was also performed with the default L-BFGS optimizer, where the convergence criterion was computed in the same way as in the microiterations scheme. The value for c in both optimizers was set to $0.53\frac{\mathrm{kcal}}{\mathrm{mol}\cdot\text{Å}}$. This corresponds to 0.00045 au, which has been recommended in literature.[109]

The default L-BFGS optimizer needs 1105 optimization steps for minimizing the energy from $-40214.3\frac{\mathrm{kcal}}{\mathrm{mol}}$ to $-46315.6\frac{\mathrm{kcal}}{\mathrm{mol}}$. The results for the optimizations with the microiteration scheme are summarized in Table 5.4. Without adjustment of charges, the optimizer runs for 22 cycles with a total of 1464 micro- and 182 macroiterations. Convergence is reached at an energy of $-46318.1\frac{\mathrm{kcal}}{\mathrm{mol}}$. So the number of QM/MM calculations decreases significantly, and the minimum geometry found was even lower in energy than the conformation obtained by the simultaneous relaxation of all atoms. When using the QM charges as charge parameters for the microiterations, convergence was reached after 23 cycles with 1611 micro- and 230 macroiterations in total at an energy of $-46329.2\frac{\mathrm{kcal}}{\mathrm{mol}}$. Although the number of QM/MM evaluations increases slightly (from 182 to 230), it is still only a fraction of that necessary with the default optimizer (1105). These additional computations lead to an energy gain of more than $10\frac{\mathrm{kcal}}{\mathrm{mol}}$.

	Force Field Parameters			QM Charges as Parameters		
Cycle	MM	QM/MM	Energy [kcal/mol]	MM	QM/MM	Energy [kcal/mol]
1	910	56	-46288.7	1082	74	-46306.6
2	308	30	-46311.0	194	45	-46318.1
3	92	8	-46314.4	153	7	-46325.0
4	11	17	-46314.9	11	8	-46325.5
5	14	4	-46315.4	14	8	-46325.9
6	7	6	-46315.6	7	3	-46326.2
7	15	3	-46316.0	44	7	-46326.9
8	6	7	-46316.2	8	3	-46327.2
9	29	3	-46316.7	8	15	-46327.5
10	8	7	-46317.0	10	4	-46327.8
11	8	3	-46317.2	14	4	-46328.0
12	8	12	-46317.4	6	3	-46328.2
13	12	4	-46317.6	2	12	-46328.2
14	15	2	-46317.8	7	4	-46328.4
15	4	8	-46318.0	9	2	-46328.5
16	6	3	-46318.1	15	7	-46328.8
17	2	1	-46318.1	6	3	-46328.9
18	3	2	-46318.1	7	12	-46329.0
19	2	2	-46318.1	7	2	-46329.1
20	2	2	-46318.1	3	3	-46329.2
21	2	2	-46318.1	2	2	-46329.2
22	0	0	-46318.1	2	2	-46329.2
23	-	-	-	0	0	-46329.2
TOTAL	1464	182	-46318.1	1611	230	-46329.2

Tabelle 5.4: Results of the optimizations with microiterations. The columns marked with "MM" and "QM/MM" give the number of microiterations and macroiterations in each optimization cycle, respectively. The third column gives the energy reached after the current cycle. The last row summarizes the total number of energy and gradient evaluations during the optimization, as well as the final energy.

5.4.2 OPT++ Optimizer for Constraint Optimization

A common method to investigate reaction paths is the scan of the PES, which can be performed with the task `2DSCAN` in CAST.[112] Two reaction coordinates are chosen, which either correspond to bonds, angles, or dihedral angles. The range to be scanned is defined as a grid on the surface spanned by these reaction coordinates. The system is moved along the grid. At each step, a local optimization is performed, where the reaction coordinates are constrained at the given point. For this purpose, an optimization method is required that allows constraining such compound coordinates which are defined by the positions of several atoms. In the original implementation, this task could exclusively be executed with the Chemshell[113] interface, using its built-in optimizer which provides the needed functionality.[112]

Now, use of the `2DSCAN` task should be enabled with any energy interface, especially the QM/MM schemes. For this reason, it must work together with one of the optimizers implemented in CAST. But neither the L-BFGS optimizer nor the TRIC one is suitable for this. In the L-BFGS implementation of CAST, it is not possible to constrain reaction coordinates like distances, angles, or dihedrals. The TRIC optimizer can in principle be used for constraining such internal coordinates and can also be combined with the `2DSCAN` task. But since it uses internal coordinates instead of cartesian ones, the gradients have to be converted from their cartesian to the internal representation in each optimization step. This leads to a very bad performance for large systems, which are usually computed with QM/MM.[83] Furthermore, it is not possible to fix atoms, a feature frequently used when computing reaction paths of enzymes.

To provide an optimizer that fulfills all the listed requirements, the library OPT++ was included in CAST.[114] It contains a variety of optimization methods which can be applied to non-linear functions. Some of them also allow for setting constraints. Out of those methods, a suitable one had to be chosen: For the problem to be optimized (i. e. the energy function), first derivatives are available (gradients), but in most cases no second derivatives (hessian matrix) or a least squares function operator. Out of all methods applicable under these conditions, only FDNIPS and QNIPS provide the possibility to set general constraints. Both are based on the Nonlinear Interior Point Method (NIPS), where the hessian is approximated by finite differences (FD) or by BFGS.[114]

General Remarks

In order to solve an optimization problem, it has to be stored in an object of one of the NLP (nonlinear problem) classes provided by OPT++.[114] There are three child classes of NLPBase, which differ in the availability of derivatives for the function: NLP0 represents a problem without any derivative information, NLP1 provides first derivatives, and NLP2 provides also second derivatives. For constructing these NLP objects, there are three corresponding NLF (nonlinear function) classes, NLF0, NLF1 and NLF2.[114] For creating the NLP object, the following information has to be given to the constructor of the NLF class:[115]

- the dimension of the problem
- a pointer to a C++ function opt_function() that evaluates the problem
- a pointer to a C++ function init_function() that initializes the problem
- optional: a constraint object and a void pointer to additional information needed by opt_function()

The newly developed routines for evaluation and initialization of the problem need to follow the interfaces defined by OPT++. A function with first derivatives has the following layout:[115]

```
void opt_function(int mode, int ndim, const NEWMAT::ColumnVector& x,
      double& fx, NEWMAT::ColumnVector& gx, int& result, void* vptr)
```

Here, mode defines if only the function value (energy) should be evaluated or if the gradients should also be computed. ndim is the dimension of the function. x is a vector of size ndim, which contains the values for all variables (coordinates). After running the function, the resulting function value for the current x is stored in fx, and the gradients are saved in gx. result contains information about the kind of evaluations available (depending on mode). Through the void pointer vptr, any additional information can be given to the function. This is very useful if the function depends on any additional parameters except x, which are not known at compile time.[115]

For the initialization function, the signature is defined as:[115]

```
void init_function(int ndim, NEWMAT::ColumnVector& x)
```

Again, ndim is the dimension of the function. x contains the reference to a vector, which will be filled with the initial values inside this function. Since no additional parameters can be given to this function, the initial values for x either have to be hard-coded or to be available globally.[115]

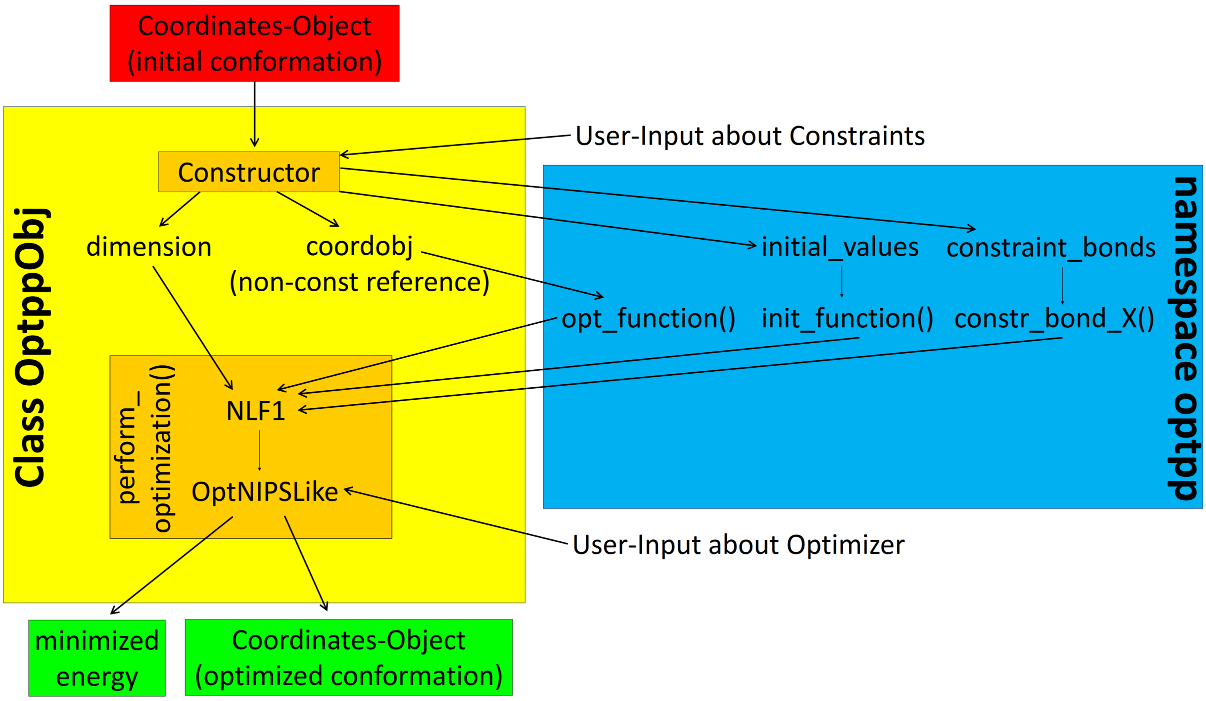


Abbildung 5.15: Overview of the implementation of the OPT++ optimizer

Implementation Details

The OPT++ optimizer was implemented in a class `OptppObj`. So the `o()` function of the `Coordinates` class only has to create an instance of `OptppObj` and call a member function of this object that performs the optimization. A schematic overview of the implemention of the OPT++ optimizer is shown in Figure 5.15.

The constructor of `OptppObj` receives an object of type `Coordinates` as argument. A reference to this object is saved in a member variable called `coordobj`. This ensures that it can be changed inside the class and that those changes also apply to the object outside. Specifically, the `this` pointer to the current `Coordinates` object is given to the constructor. The energy is minimized inside `OptppObj` and the optimized atom positions are saved.

Furthermore, the constructor calculates the dimension of the problem as $3N$, where N is the number of atoms. The result is stored in a member variable `dimension`. The atomic coordinates before starting the optimization are converted to a `NEWMAT::ColumnVector` in the order x1, y1, z1, x2, y2,... Since they are needed inside the initialization function, the resulting vector is saved as a global variable called `initial_values`. To ensure that the variable is not accidently used outside the `OptppObj` class, it is defined in a namespace `optpp` with the keyword `static`. This limits the accessibility to the current translation unit.

Afterwards, the member function `perform_optimization()` is called. It constructs the `NLF1` object from the functions `init_function()` and `opt_function()` and runs the optimization. Those two functions are both defined in the namespace `optpp`.

The initialization function sets `x` to `initial_values`. `opt_function()` receives a void pointer to the current `OptppObj` as argument in order to access its member variable `coordobj`. It fills the given `x` values into the `Coordinates` object as atomic positions. Then it calls the `g()` function. If `mode` requests the function value, `fx` is set to the resulting energy. If the gradients are also required, they are converted to a `NEWMAT::ColumnVector` and stored in `gx`.

For the optimization, a unique pointer to `OptNIPSLike` is created, which is the common base class of `OptQNIPS` and `OptFDNIPS`.[114] According to the user's wishes, a specific object of either `OptQNIPS` or `OptFDNIPS` is created and linked to the unique pointer. Then the optimizer is configured with the user-given options from the CAST input file, which correspond to the algorithmic parameters in OPT++.[115] After running the member function `optimize()` of the chosen optimizer, the vector `x` is converted to atomic positions which are saved in `coordobj`. The function value, i. e. the minimized energy, is returned.

Distance Constraints

In OPT++, it is possible to set different kinds of constraints, specifically bound constraints (i. e. a lower and an upper bound for the variables), linear constraints, and nonlinear constraints.[114] Since for the scan of the PES it is necessary to fix internal coordinates, which depend nonlinearly on the cartesian ones, only the last type of constraints is needed. The corresponding class, `NonLinearEquation`, in which such a constraint is stored, inherits from `NonLinearConstraint`. This `NonLinearConstraint` contains a pointer to an NLP object, which has to be created as explained before. Instead of the function to be optimized (`opt_function()`), the a constraint function $f(x)$ has to be provided. It describes the constraint in a way that $f(x) = 0$ should always be true during the optimization.[114] The signature of this function is:[115]

```
void constr_function(int mode, int ndim, const NEWMAT::ColumnVector& x,
           NEWMAT::ColumnVector& cx, NEWMAT::Matrix& cgx, int& result)
```

The meaning of `mode`, `ndim`, `x`, and `result` is the same as for the `opt_function()`. The function value is saved in the variable `cx`, and the gradients in `cgx`. They have the format of a vector and a matrix respectively in order to combine several constraint conditions in one function.[115] This feature is, however, not used in the current implementation in CAST. Only the first element of the vector and the first column of the matrix are filled.

Instead, a separate function is defined for each constraint. This way, the number of constraints is limited, at the moment to two atomic distances. The function $f(x)$ for two atoms A and B, whose distance is fixed to d_{AB}, can be written as:

$$f(x) = |R_A - R_B| - d_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} - d_{AB} \tag{5.21}$$

In order to evaluate this function and its derivatives, the positions of atoms A and B as well as the distance d_{AB} have to be known. As the constraint function does not accept any additional parameters, this information has to be available globally. For this reason, the namespace `optpp` contains another static variable `constraint_bonds`, which is a vector of self-defined structs called `constraint_bond`. Each of these structs holds d_{AB}, which is given by the user, and the six indices of the required coordinates in vector `x`. They can be derived directly from the user-given atom indices i through $3i + n$, with $n = 1$ for the x-coordinate, $n = 2$ for the y-coordinate, and $n = 3$ for the z-coordinate of atom i.

There are two constraint functions, `constr_bond_1()` and `constr_bond_2()`, whose only difference is that they use the first and the second element of `constraint_bonds` respectively. With the help of the information stored therein, they extract the coordinates (x_A, y_A, z_A, x_B, y_B, z_B) from the input vector `x`. Then they calculate the function value according to Equation 5.21 and the derivatives of this function with respect to the atomic coordinates. From `constr_bond_1()` and `constr_bond_2()`, two constraint objects of type `NonLinearConstraint` are created. Both are combined into a `CompoundConstraint`, which is an object that contains an array of different constraints.[114] This compound constraint is applied to the NLF of the optimization problem.

Fixation of Atoms

Sometimes, especially for big molecular systems, not all atoms should be moved during an energy minimization. The option to fix some of the atoms is implemented in the OPT++ optimizer by excluding the coordinates of the fixed atoms from the variables of the optimization problem. Specifically, the following changes have to be made:

- When computing the dimension of the problem in the `OptppObj` constructor, only the non-fixed atoms have to be taken into account. This ensures that the dimension of the problem decreases, which leads to an improved performance.

- For conversions from the `Coordinates` object to a `NEWMAT::ColumnVector`, only the values of the non-fixed atoms have to be included. This holds true for the atomic positions when creating the `initial_values` vector, and for saving the gradients into `gx` during `opt_function()`.

- When filling atomic positions contained in a `NEWMAT::ColumnVector` into the `Coordinates` object, attention has to be paid to fill only the elements of the non-fixed atoms with their corresponding values. This is done once in `opt_function()` before evaluating energy and gradients, and once at the end of `perform_optimization()` to save the final geometry.

When running an optimization with fixed atoms and distance constraints applied on non-fixed atoms, the indices to be saved in the `constraint_bond` objects must refer to a vector consisting only of the coordinates of the non-fixed atoms. Thus, they cannot be calculated directly from atom indices i of the involved atoms as described above, because this would yield the indices in a vector consisting of all atomic coordinates (including the fixed ones). Instead, i has to be defined as the index of the corresponding atom in a list consisting only of the non-fixed atoms. Then, the above formula $3i + n$ can be applied to find the values to be stored.

6 Investigation of the Reaction Path of Rhodesain and K11777

To demonstrate the possibilities of the novel QM/MM methods in CAST, some investigations of the reaction path for the complex formation between rhodesain and the inhibitor K11777 were performed. These include singlepoint calculations, 2D scans, molecular dynamics simulations, and Umbrella Samplings.

Rhodesain, also called Cathepsin L or Brucipain, is a Clan CA, family C1 (papain-family) cathepsin L-like cysteine protease, expressed in *Trypanosoma brucei rhodesiense*, a parasite causing Human African Trypanosomiasis (HAT).[116]

This disease, which is better known as sleeping sickness, comes in two stages:[117] During the first stage, the parasites live in subcutaneous tissues, blood, and lymph, where they cause unspecific symptoms like fever, headache, muscle pain, and enlarged lymph nodes. When the trypanosomes cross the blood-brain barrier, the second stage begins. Subsequently, the typical symptoms appear, like mental deterioration and disturbances of the sleep cycle. Without treatment, HAT is considered fatal: The patient will die in the end.[117]

Although there has been significant success in fighting this disease (cases have dropped from around 10000 to 977 between 2008 and 2018), it still threatens millions of people in sub-Saharan Africa.[117] Drugs used for the treatment of HAT patients, especially in the second stage, are often only effective against *T. b. gambiense*.[117] This is another subspecies of *Trypanosoma*, which is more widespread than *T. b. rhodesiense* (it causes 98% of the currently reported cases), but the infection shows a slower progression and a lower mortality rate.[116,117] The first-line treatment for *T. b. rhodesiense* is Melarsoprol, which has severe side-effects, like encephalopatic syndrome with a mortality of 3 to 10%.[117] Therefore, new approaches for treating this disease are still urgently in need.

An attractive target for the development of new therapeutic agents might be rhodesain, the major cysteine protease in *T. b. rhodesiense*.[118] Previous studies in mice showed that suppression of this enzyme prolonged the lifespan of the infected animals.[119] Investigations with an *in-vitro* model of the blood-brain barrier suggest that this might be due to the inability of the parasites to enter the central nervous system, where they induce the lethal second stage of the disease.[119]

Rhodesain has the typical two-domain fold structure of papain-family cysteine proteases (see Figure 6.1).[120] The active site is located in the cleft between the left and the right domain. This catalytic dyad consists of the sidechains of the residues Cys25 and His162.[120] Cysteine is deprotonated while histidine is protonated, leading to an increased nucleophilicity of the sulfur atom.[121]

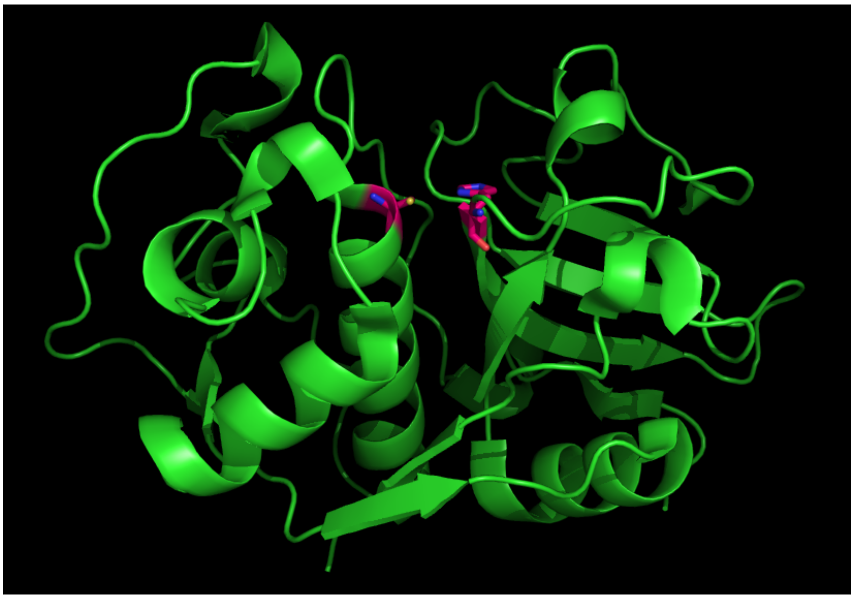


Abbildung 6.1: Total Structure of rhodesain with the catalytic dyad marked in red

A substance known to act as an inhibitor for rhodesain is K11777.[120] Its warhead is a vinylsulfone that binds irreversibly to the catalytic dyad of the protease.[120] The mechanism is shown in figure 6.2. In the first step, the deprotonated sulfur atom of cysteine attacks the double bond of the inhibitor. Then the proton of the histidine residue is transferred to the other carbon atom of the former double bond.[121] The resulting complex is stabilized by a network of polar interactions (see figure 6.3).[120] A crystal structure of this complex as well as kinetic data for its formation are available.[120]

K11777 is an interesting candidate not only because of its reactivity with rhodesain. It has also entered pre-clinical trials against Chargas disease, another form of trypanosomiasis which is endemic in many countries of Central and South America.[116,122] Furthermore, the inhibitor showed promising activity against several viruses in cell cultures, including the coronaviruses SARS and MERS.[123] For this reason, it might also be an agent against SARS-CoV-2, which caused a global pandemic in 2020 leading to a shutdown of public life in many countries.[124]

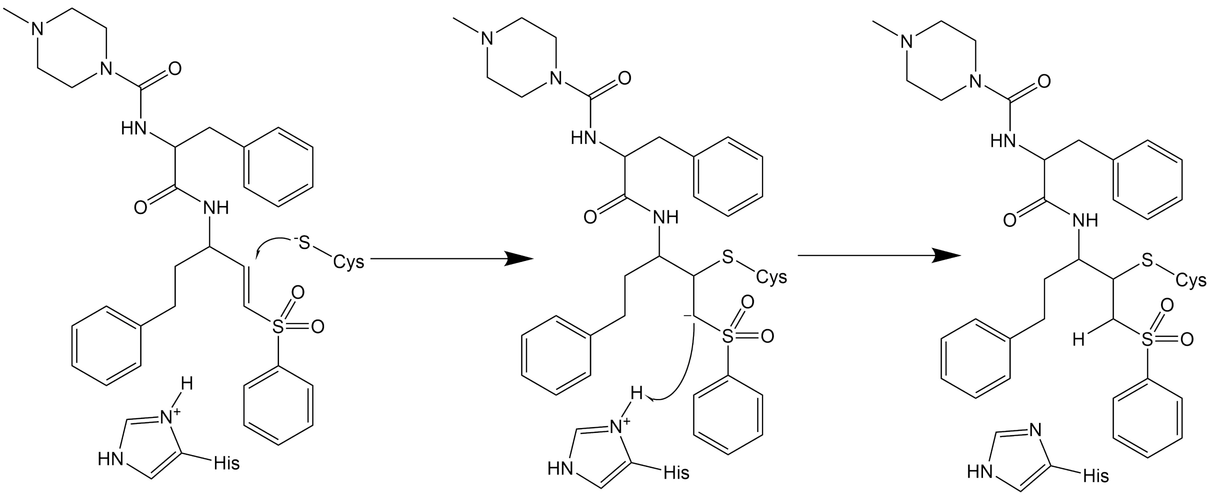


Abbildung 6.2: Mechanism for the formation of the rhodesain-K11777 complex

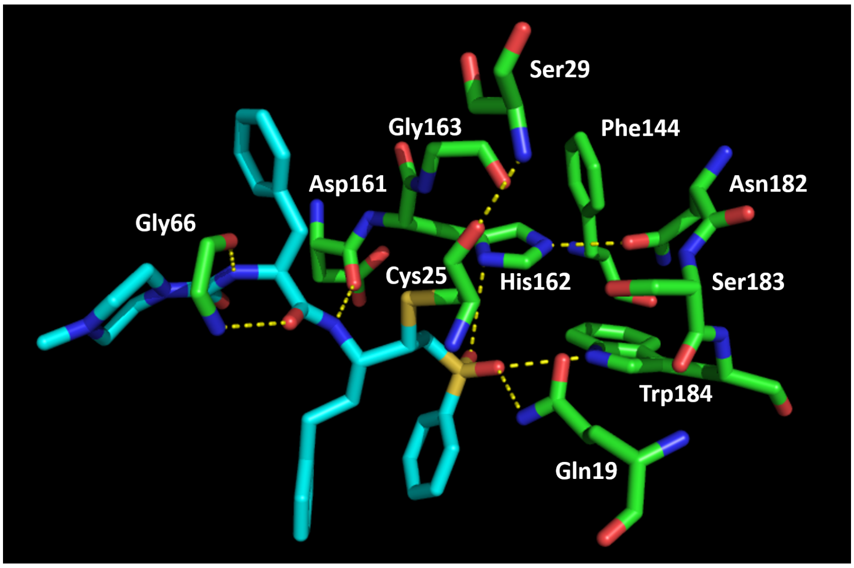


Abbildung 6.3: Active site of rhodesain with bound inhibitor K11777

6.1 Benchmark of the QM/MM Methods

In order to benchmark the different options for QM/MM in CAST, singlepoint energies were evaluated for a number of structures along the reaction path. The structures were provided by Waldemar Waigel.[125] To obtain them, an MD of the non-bonded enzyme inhibitor complex was run, i. e. there was no covalent bond between the cysteine sulfur and the inhibitor. Then the system was cut to a sphere of 12 Å around the catalytic dyad using the TAO toolkit.[126] The following 2D scan was performed with ONIOM, where only an inner sphere of 6 Å was allowed to move during the optimizations. The QM region consisted of 43 atoms (*small*, see Figure A.1) and was treated with ωB97XD/6-31+G*. The rest of the system was described by the AMBER force field. The structures that form the lowest energy path from the non-covalent to the covalent complex were chosen from the resulting surface, which is spanned by the S-C distance corresponding to the enzyme inhibitor bond and the distance between the proton and the histidine nitrogen atom.[125]

6.1.1 Comparison to Gaussian-ONIOM

At first, the general performance of the multi-layer methods in CAST was validated. This was done by a direct comparison to the ONIOM implementation of Gaussian. The reference path was taken from Jessica Meyr's bachelor thesis.[127] It was obtained using a three-layer scheme, where the innermost layer consisted of 43 atoms (*small*) and the intermediate layer of 89 atoms, leading to a *medium* system of 132 atoms called *full* (see Figure A.1) Both inner layers were computed with B3LYP, where 6-31+G(d) was used as the basisset for the inner layer and 6-31G for the intermediate one. The remaining 3237 atoms of the structure were treated with the AMBER force field that is implemented in Gaussian.[127] In CAST, Psi4[57] was applied as interface for the inner layer and Gaussian[56] for the buffer zone. In order to mimic the behavior of ONIOM, the external charges for the *medium* system were determined by option *delM3* (see section 4.1.2) and those for the *small* system with EE (see section 4.2.1). This means that all MM atoms, which are separated from the *medium* system by more than three bonds, are considered for the creation of external charges in both model systems. The outer layer was treated by a force field implemented in CAST. Two paths were calculated, one with the AMBER and one with the OPLSAA force field applied as the MM interface.

The results are depicted in Figure 6.4. The solid lines represent the overall three-layer energies. The dashed and the dotted lines show the QM and MM contributions respectively, i. e. $E_{small,QM}$ and $E_{big,MM} - E_{small,MM}$ in Equation 4.15. Both paths cal-

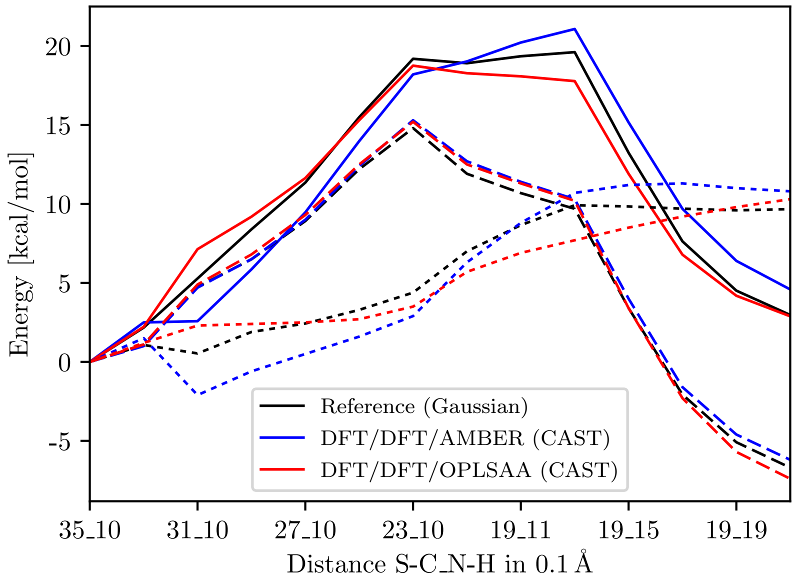


Abbildung 6.4: Comparison of the reaction paths calculated with Gaussian and CAST using the three-layer interface (dashed lines: only QM contribution, dotted line: only MM contribution)

culated with CAST are quite similar to the reference path, with a maximum deviation of less than $5\frac{\text{kcal}}{\text{mol}}$. While the QM contribution is nearly identical, bigger differences are found in the MM contribution. This is due to the fact that the force field atom types and parameters used in Gaussian are different from those in CAST, even if both programs use the AMBER force field. Since the relative energies computed with OPLSAA are closer to the reference than those obtained with AMBER, OPLSAA will be used for the following calculations.

6.1.2 Influence of Different Options

For performance reasons, the benchmark of different QM/MM options should be done with DFTB3 from the DFTB$^+$ interface instead of DFT. In order to test how this influences the paths, further three-layer calculations were performed, where either the inner layer or the intermediate layer was computed with DFTB3. Another path was obtained using the subtractive QM/MM interface with QM system *full*, i. e. both inner layers of the former three-layer scheme were treated with DFTB3.

Figure 6.5 shows the resulting paths. If the interface for the intermediate layer is replaced by DFTB3, the shape of the curve remains more or less unchanged. Only the relative energy of the covalent complex increases a little from $2.8\frac{\text{kcal}}{\text{mol}}$ to $6.3\frac{\text{kcal}}{\text{mol}}$. If, however,

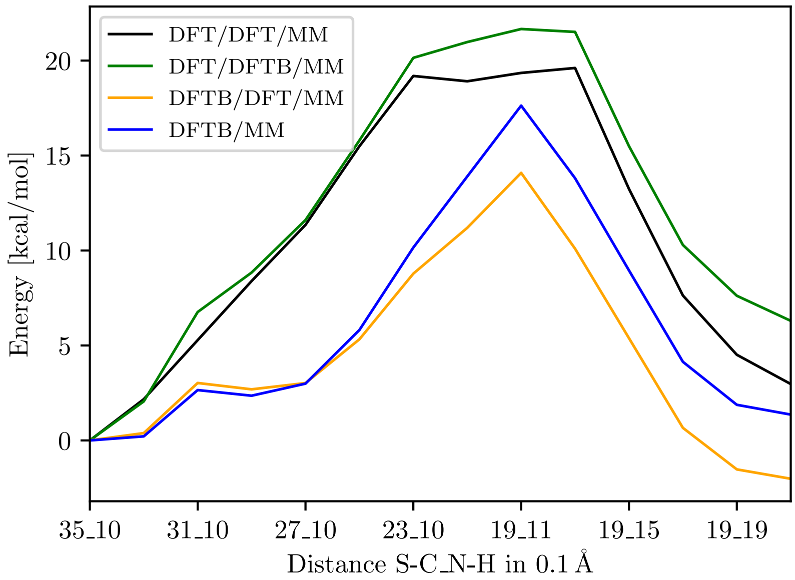


Abbildung 6.5: Comparison of the reaction paths calculated with DFTB3 as interface either for the innermost or the intermediate region or for both inner layers

the active site is computed with DFTB3, the shape of the path changes significantly, irrespective of whether there is an additional buffer layer or not. The transition state is a peak at PES point 19_11 (this means that the S-C distance is 1.9 Å and the N-H distance is 1.1 Å), whereas it forms a plateau between PES points 23_10 and 19_13 when the active site is described by DFT. Computing the intermediate layer with DFTB3 instead of DFT again causes the relative energy of the bonded state to increase about 3.5$\frac{\text{kcal}}{\text{mol}}$, from $-2.0\frac{\text{kcal}}{\text{mol}}$ to 1.4$\frac{\text{kcal}}{\text{mol}}$.

Although DFTB3 seems to fail in giving paths similar to DFT, it can still be used for testing the different options of the QM/MM interface. The aim of QM/MM is to approximate the QM description of the total system by only applying the QM method to a small subsystem, which is embedded in an environment described by the MM method. For a benchmark, all QM/MM calculations should therefore be compared to a reference obtained by computations of the total system with the QM method. Evaluating the energy of a system consisting of several thousands of atoms with a relatively high-level DFT method like B3LYP is computationally infeasible. But it can be done using a semi-empirical method like DFTB3, which provides a suitable reference for the following benchmarks.

Additive vs. Subtractive QM/MM

The first question to be investigated is whether the additive or the subtractive QM/MM scheme is more successful in reproducing the path where the total system is computed with DFTB3. The inner region for the two-layer calculations was the QM system *full* (see Figure A.1). As described above, the external charges were created from the MM atoms which exhibit a distance to the QM region of more than three bonds (*delM3*). The OPLSAA force field was applied as interface for the outer layer.

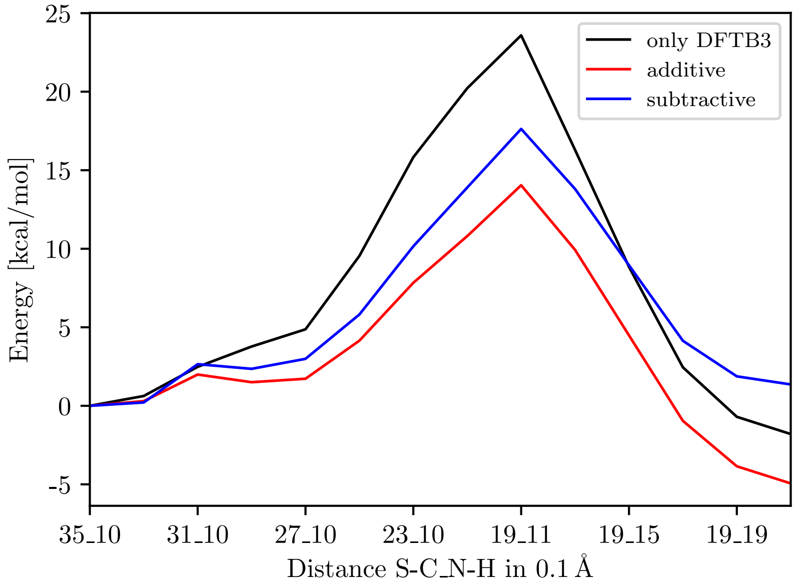


Abbildung 6.6: Comparison of the reaction paths calculated with additive and subtractive QM/MM

The relative energies along the paths are depicted in Figure 6.6. It is apparent that the general shape of the curves is identical for all three paths, with a peak at PES point 19_11. This is consistent with the findings above, leading to the conclusion that the shape depends on the method used for the active site, while it is less sensitive to the method chosen for the rest of the system. The deviation of the reaction energy in comparison to the pure DFTB3 calculation is around $3\frac{\mathrm{kcal}}{\mathrm{mol}}$ for both QM/MM paths. While the reaction is described as more exothermic with the additive scheme, the subtractive scheme predicts a slightly endothermic reaction. The relative energy of the transition state with respect to the non-covalent complex is lower in the frame of QM/MM than in the reference path. Using the subtractive scheme, it decreases by around $6\frac{\mathrm{kcal}}{\mathrm{mol}}$, in the additive scheme by nearly $10\frac{\mathrm{kcal}}{\mathrm{mol}}$.

To sum it up, the differences between the additive and the subtractive QM/MM interface are relatively small. For the rest of the benchmark, the subtractive scheme will be used. The reason is not only the better description of the transition state observed in Figure 6.6, but also the better comparability to the three-layer scheme which is also tested.

Embedding Scheme (Two-Layer)

The next aspect to be tested was the embedding scheme in the two-layer interface, i. e. which MM atoms should be taken into account when creating the external point charges for the QM system. Calculations were performed applying electrostatic embedding with the options *delM1*, *delM2*, and *delM3*. They differ in the number of bonds by which an MM atom must be separated from the nearest QM atom in order to be considered for the creation of the external charges (see section 4.1.2). Furthermore, a path was obtained using mechanical embedding, where no external charges are added to the QM system at all.

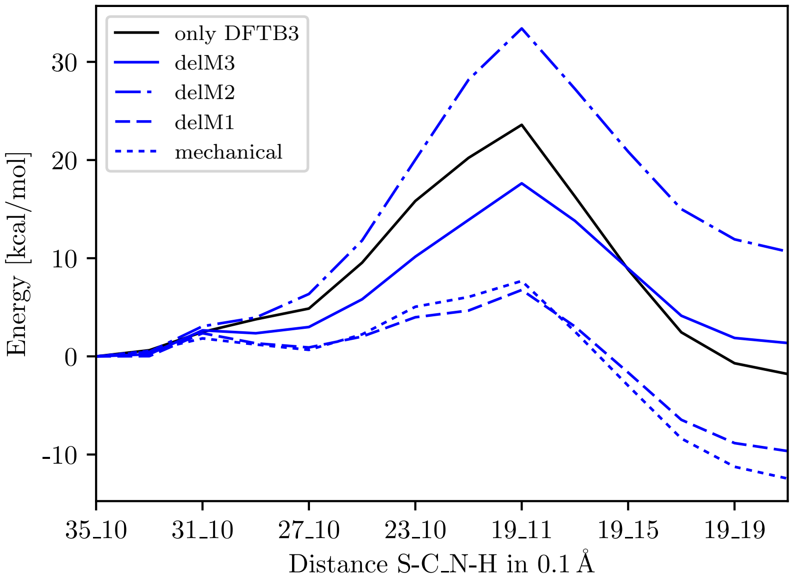


Abbildung 6.7: Comparison of the reaction paths calculated with different embedding schemes in subtractive QM/MM

From Figure 6.7, it can clearly be seen that *delM3* gives the best relative energies, defined as most similar to the path computed using only DFTB3. This option is identical to the default setting *scalecharge=500* in Gaussian-ONIOM.[56] The paths obtained with *delM1* and *delM2* show deviations of around $10\frac{\text{kcal}}{\text{mol}}$ from the reference, where the relative energy of transition and final state is lower for *delM1* and higher for *delM2*. The path computed with mechanical embedding is most similar to the one calculated using *delM1*. This is surprising, because *delM1* includes a maximum of external charges into the QM calculation, while no external charges are included at all when mechanical embedding is applied. It seems that the error made by evaluating the QM/MM interactions on the MM level (mechanical embedding) is about the same size as the one from the overcounting of interactions with *delM1*.

Embedding Scheme for Inner System (Three-Layer)

After the investigation of the embedding scheme for two-layer calculations, the influence of the embedding scheme in three-layer computations should be studied. Since the creation of external charges for the *medium* system is identical to the two-layer scheme, only the charges for the *small* system are treated here. As described in section 4.2.1, there are four different options regarding them: EEx, EE, EE+, and EE+X. EEx does not include any external charges into the computations of the *small* system. With EE, the charges used for the *small* system are the same as those for the *medium* one. If EE+ is chosen, even charges from atoms of the outermost layer are taken into account that are not considered for the *medium* system, but the atoms of the intermediate layer are ignored. Only when selecting EE+X, atoms from both outer layers are used for creating the external charges for the *small* system.

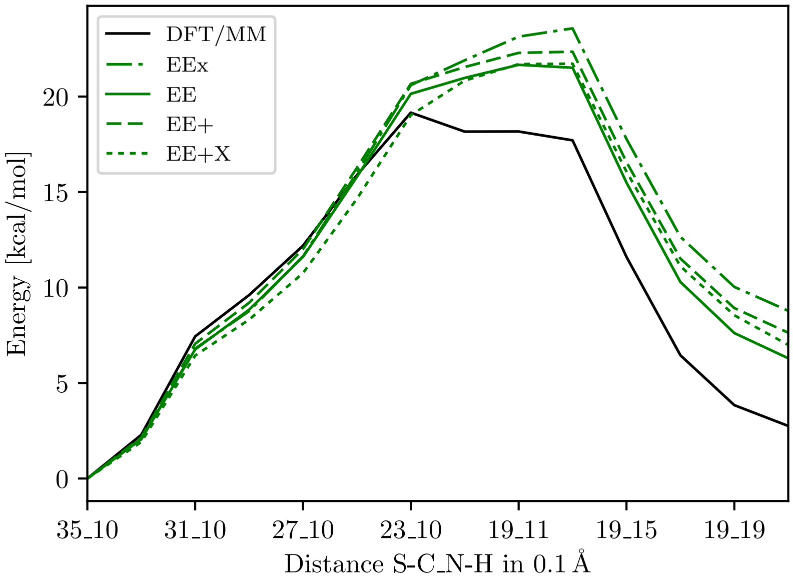


Abbildung 6.8: Comparison of the reaction paths calculated with different embedding schemes in three-layer scheme

The partition of the total system was the same as in the original computation with 43 atoms in the inner layer, 89 atoms in the intermediate layer, and 3237 atoms in the outer layer. The *small* system was treated with B3LYP/6-31+G(d) from the Gaussian interface, the buffer layer was computed with DFTB3. A path obtained by the subtractive QM/MM interface served as reference, where all 132 atoms of QM system *full* were calculated with B3LYP/6-31+G(d).

The results are shown in Figure 6.8. The paths computed with the three-layer interface are very similar to each other, with a maximum distance in reaction energy of $2.5\frac{\mathrm{kcal}}{\mathrm{mol}}$ between EE and EEx. In comparison to the reference path, the reaction energy is around $5\frac{\mathrm{kcal}}{\mathrm{mol}}$ higher, with the smallest deviation to option EE and the largest to EEx. Therefore, including MM charges into the computations of the innermost layer seems to improve the description of the interactions slightly, but apparently they do not have a big influence on the relative energies.

6.1.3 Influence of the QM Region

Another important aspect influencing the behavior of QM/MM calculations is the choice of the QM region. A number of investigations considering this point has been performed before in the frame of three-layer calculations, suggesting some promising QM regions.[127–129] Two of these are compared in the following. An overview of which atoms are included in each can be found in the appendix, Figure A.1.

QM system *full* was developed by Jessica Meyr.[127] In addition to the warhead of the inhibitor and the residues of the catalytic dyad (Cys25 and His162), it also contains an extended backbone of these two amino acids, as well as the residues of some amino acids in spatial proximity to the active site. These are Ser29, Phe144, Ser183, Gln19, Trp184, and Asn182 (see also Figure 6.3).[127] Jonas Weiser and Maximilian Asbach created QM system *new*.[128,129] It was inspired by the crystal structure, showing a network of polar interactions between rhodesain and K11777 (see Figure 6.3).[120] Like the *full* system, it contains the warhead of the inhibitor and the amino acids of the catalytic dyad with an extended backbone, as well as the side chains of Gln19, Trp184, and Asn182. Furthermore, the backbones of Asp161 and Gly66 are included. In total, QM system *new* consists of 117 atoms.[128]

Based on the findings of section 6.1.2, the calculations were performed using the subtractive QM/MM interface in CAST. The QM region was treated with DFTB3, the MM region with the OPLSAA force field. For the evaluation of the QM/MM interactions, electrostatic embedding with option *delM3* was applied.

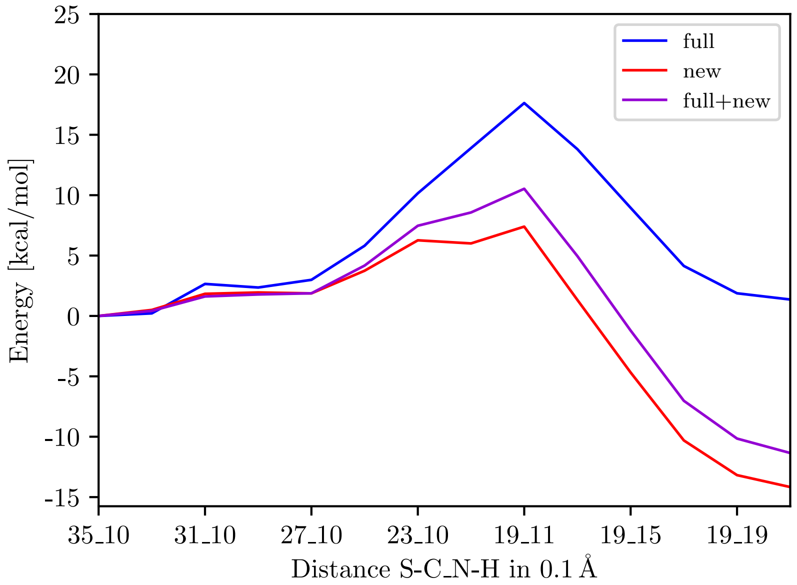


Abbildung 6.9: Comparison of the reaction paths calculated with different QM regions using structures from non-covalent path

The resulting paths are depicted in Figure 6.9. They differ significantly depending on the QM system. While the reaction is slightly endothermic with *full* ($\Delta E = 1.4\frac{\text{kcal}}{\text{mol}}$), it becomes exothermic when QM system *new* is applied ($\Delta E = -14.2\frac{\text{kcal}}{\text{mol}}$). Extending the QM region to *full+new*, containing all 154 atoms of both systems *full* and *new*, leads to an exothermic path very similar to the one obtained with *new*. These results are in better agreement with experiments, where K11777 binds irreversibly to rhodesain.[120] This suggests a negative reaction energy for the formation of the complex.

All paths shown in Figure 6.4 - 6.9 were created by singlepoint calculations on structures that were obtained from a scan starting from the non-covalent complex. These paths will be called *ncov* paths. Another scan, starting from the covalent instead of the non-covalent complex, was performed by Waldemar Waigel. Apart from the starting point of the MD simulation, the procedure for obtaining structures was the same as described in the beginning of this section (see page 110). Singlepoint energies on the resulting structures were computed with CAST, using the same settings as for the *ncov* paths, in order to obtain the so-called *cov* paths shown in Figure 6.10.

As in the *ncov* calculations, the relative energy of the non-covalent end of the path increases when QM system *new* is applied instead of *full*, i. e. the reaction becomes more exothermic. The energies computed with *new* and *full+new* are nearly identical. But even

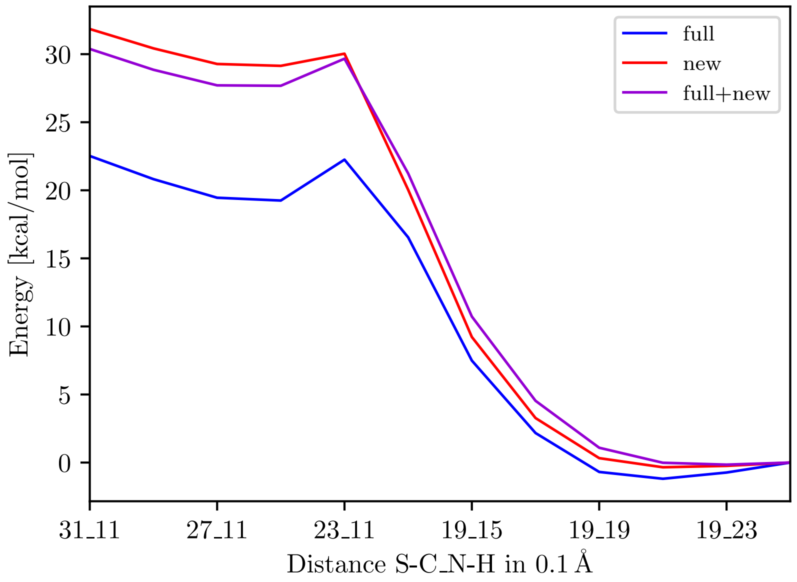


Abbildung 6.10: Comparison of the reaction paths calculated with different QM regions using structures from covalent path

more significant than the difference between QM regions *full* and *new* is the difference between the *ncov* and the *cov* paths. The reaction energy changes from $1.4\frac{\text{kcal}}{\text{mol}}$ to $-22.5\frac{\text{kcal}}{\text{mol}}$ for QM system *full* and from $-14.2\frac{\text{kcal}}{\text{mol}}$ to $-31.8\frac{\text{kcal}}{\text{mol}}$ for QM system *new*. Whereas the *ncov* paths have a transition state that is higher in energy than both endpoints, the highest point in the *cov* paths is located at the non-covalent complex. This raises the question, why the paths are so different and which of them approximates the real energy change during the reaction best.

This question has also been studied in previous investigations.[127–129] It was found that the conformation of the enzyme is different, depending on the starting structure.[128] If the inhibitor is covalently bound, the substrate pocket of the enzyme is open, whereas it is closed in case of a non-covalent complex. As this does not change significantly during the scan, it might account for the observed differences in energy.[128] This is illustrated in Figure 6.11. The *ncov* path corresponds to a reaction in the closed state, where the overall energy change is nearly zero. On the other hand, the enzyme is in the open conformation during the *cov* path, so the energy decreases while the covalent complex is formed.

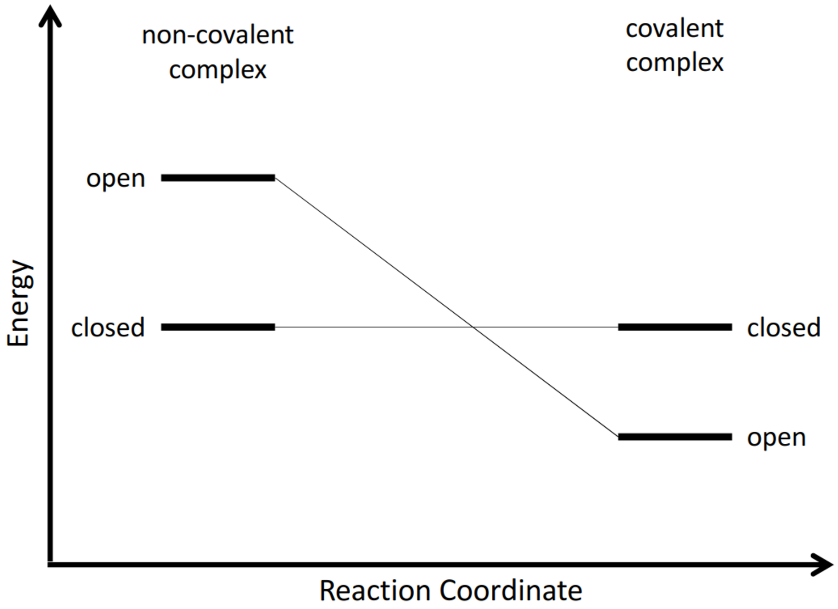


Abbildung 6.11: Influence of the conformation of the substrate pocket on the energy change along the path

6.2 2D Scans

Since the differences in the paths are caused by different conformations of the structures involved, singlepoint calculations will not solve this issue, no matter which settings are applied for the energy evaluation. Instead, the 2D scans, which had been performed with the minimal QM region *small* consisting only of the warhead of the inhibitor and the side chains of the catalytic dyad, should be repeated with the QM systems *full* and *new* described in section 6.1.3. This was done in order to see if a bigger QM region leads to a better adjustment of the substrate pocket along the reaction path.

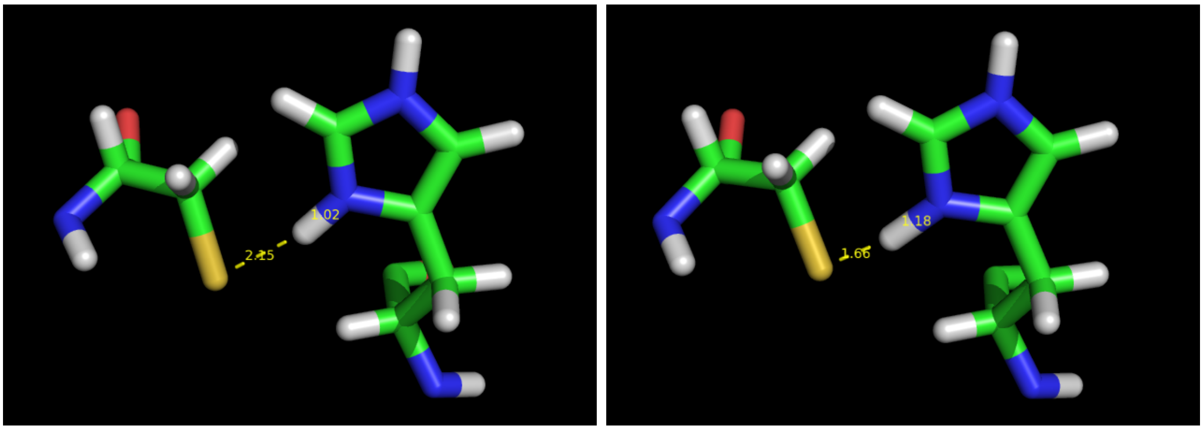
All computations in this section have been performed using the additive QM/MM interface, where the QM system was treated with DFTB3 and the rest of the structure with the OPLSAA force field. The QM/MM interactions were evaluated using electrostatic embedding with the option *delM3*.

6.2.1 Local Optimizations

To obtain suitable starting structures for the 2D scans, the energy of the original structures (PES point 19_25 for the *cov* path and 35_10 for *ncov*) was minimized, both with QM region *full* and with QM region *new*. The OPT++ optimizer was used for this purpose, ensuring that the results are consistent with those that will be obtained from the scans, where the optimization in every gridpoint can only be done by OPT++ (see section

5.4.2). During the minimization, the same atoms were fixed as in the original scan, i. e. only amino acids inside a 6 Å sphere around the catalytic dyad were allowed to move.

The structure of the covalent complex did not change significantly during the optimizations. The S-C distance remained at 1.9 Å, while the N-H distance decreased slightly from 2.5 to 2.2 Å in both calculations. In the minimization of the non-covalent complex with QM system *full*, there were no significant changes either: The distance between nitrogen and hydrogen remained constant and the one between sulfur and carbon decreased from 3.5 to 3.4 Å. When the structure was optimized using QM region *new*, however, the proton started to move towards the sulfur atom of the cysteine residue, leading to an increased N-H distance of 1.2 Å (see Figure 6.12). This might suggest that the zwitterionic state of the enzyme is not that stable compared to the uncharged isomer within the description by QM system *new*.



(a) before optimization

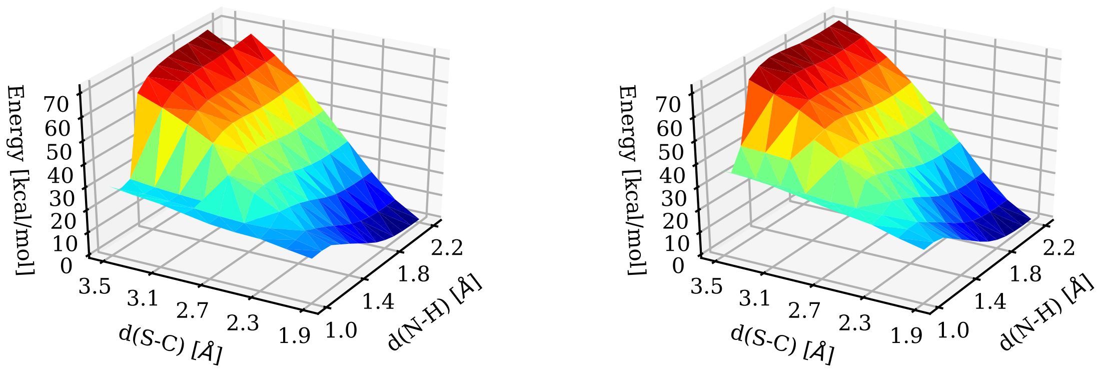
(b) after optimization

Abbildung 6.12: Movement of the proton during the optimization of the non-covalent complex with QM system *new*

Based on the results of the optimizations, it was decided to scan the S-C distance from 1.9 to 3.5 Å and the N-H distance from 1.0 to 2.2 Å. As the starting structure for the *ncov* scan with QM system *new* is not exactly located on the starting point of the PES, some caution has to be exercised regarding the values in its vicinity. The stepsize was 0.2 Å in the S-C direction and 0.1 Å in the N-H direction.

6.2.2 2D Scans from Covalent Complex

The surfaces obtained by the scans starting from the covalent complex are depicted in Figure 6.13 (the corresponding data can be found in the appendix, Table A.1 and A.2). As expected from the proposed mechanism (see Figure 6.2), the lowest energy path goes along the front edge of the surface in both scans: First the S-C distance decreases, before the N-H distance starts to increase. The only difference to the original path, calculated with the minimal QM region *small*, is that the N-H distance decreases from 1.1 to 1.0 Å intermediately.



(a) QM system *full*

(b) QM system *new*

Abbildung 6.13: Surfaces obtained by 2D scans starting from the covalent complex

The lowest energy paths, as well as the energies on the surfaces following the original path are shown in Figure 6.14. There are only small differences in comparison to the paths from the singlepoint calculations. The local maximum at step 5 (23_11), which marked the transition state, disappears completely for QM system *new* and in the lowest energy path with QM system *full*, leading to a continuously decreasing energy during the course of the reaction. The total energy change remains $-31.8\frac{\text{kcal}}{\text{mol}}$ for QM region *new* and decreases slightly from $-22.5\frac{\text{kcal}}{\text{mol}}$ to $-25\frac{\text{kcal}}{\text{mol}}$ for QM region *full*.

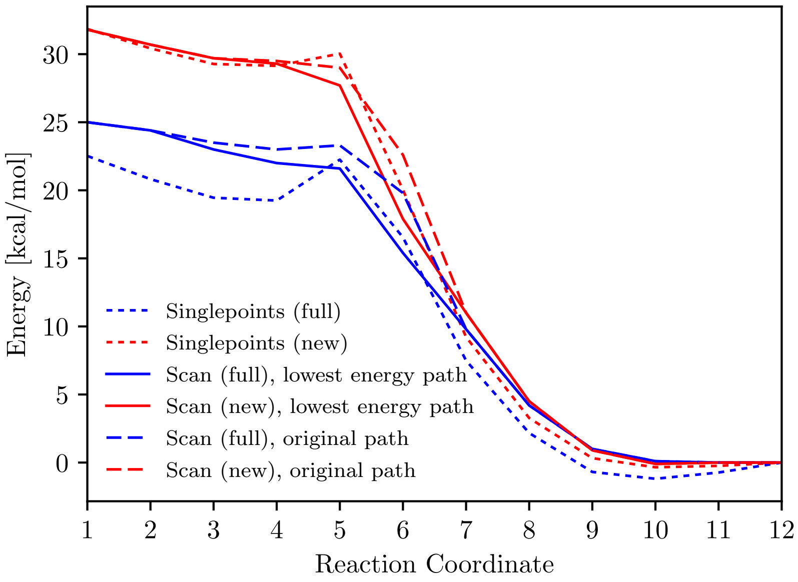


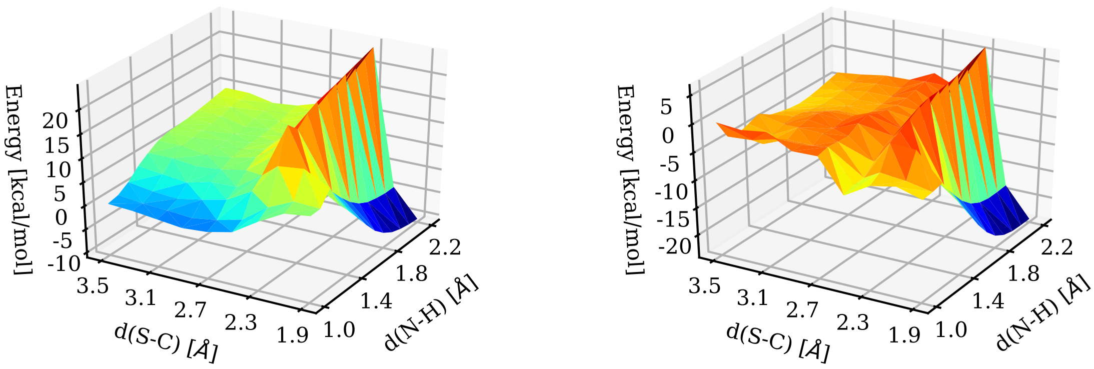
Abbildung 6.14: Reaction paths obtained by 2D scans starting from the covalent complex

6.2.3 2D Scans from Non-Covalent Complex

Figure 6.15 shows the surfaces, which were obtained by the scans starting from the non-covalent complex (data can be found in Table A.3 and A.4). It is obvious that they are quite different from those obtained from the *cov* scans (see Figure 6.13). The energy differences on the surface are smaller. While the maximum difference in energy is around $70\frac{\text{kcal}}{\text{mol}}$ in the *cov* scans, it is only around $30\frac{\text{kcal}}{\text{mol}}$ in the *ncov* scans. The shape of the surface differs, too: In the *cov* scans, the energy increases when moving away from the path (i. e. going towards the back left corner of the plot), whereas in the *ncov* scans, there is one sharp edge at d(S-C) $= 2.3\,\text{Å}$ with a relatively flat area to the left of it.

These properties are the same, independently of the choice of the QM region. In contrast to the *cov* scans, however, there are also some differences between the two calculations.

The energy of the flat area for big S-C distances is generally lower if QM system *full* is applied, leading to a less exothermic reaction. In this case, this area also descends to the front, i. e. shorter N-H distances are preferred. As a consequence, the course of the lowest energy path is exactly the same as in the original scan. It starts at PES point 35_10 (front left corner). First the Cys-S attacks the carbon-carbon bond of the inhibitor (decreasing d(S-C), moving along the front edge of the plot) and then the proton migrates from His-N to the other carbon atom of the former double bond (increasing d(N-H), moving to the



(a) QM system *full*

(b) QM system *new*

Abbildung 6.15: Surfaces obtained by 2D scans starting from the non-covalent complex

back at the right edge of the plot).

If the scan is performed with QM region *new*, however, the minimum of the left half of the surface is at PES point 35_18. Therefore, the most stable conformation for the non-covalent complex is not the zwitterionic state of the enzyme. Instead, the proton of the histidine is bound to the cysteine sulfur atom, leading to an N-H distance of 1.8 Å and a neutral catalytic dyad (see Figure 6.16). This is in agreement with the optimization described in section 6.2.1, where the proton also approximates Cys-S when the energy minimization is performed using QM system *new*. The neutral minimum energy conformation also influences the course of the lowest energy path. As a consequence, the

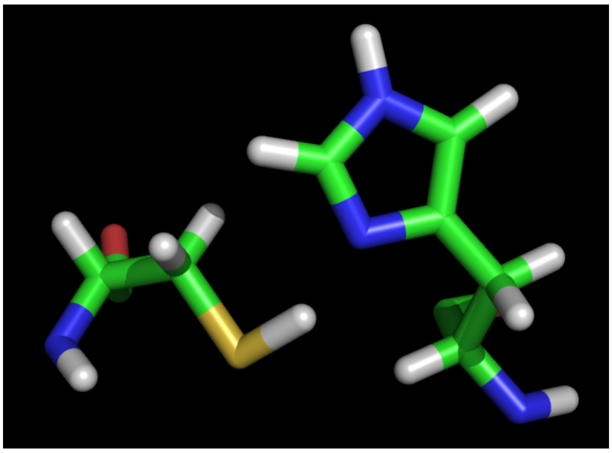


Abbildung 6.16: Minimum energy conformation for non-covalent complex, obtained from the scan with QM region *new*

reaction consists of three steps: Before the cysteine sulfur can initiate its nucleophilic attack, the proton needs to be transferred to the histidine. After the formation of the S-C bond, it is passed on to the inhibitor.

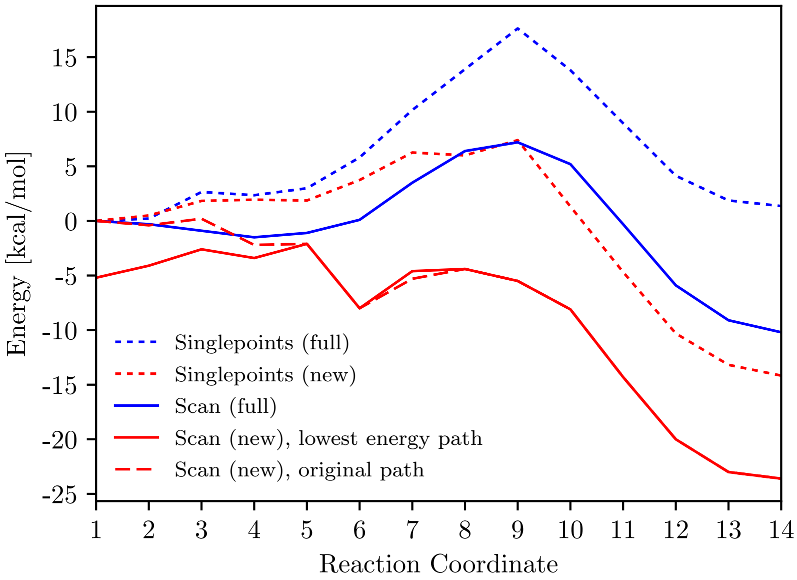
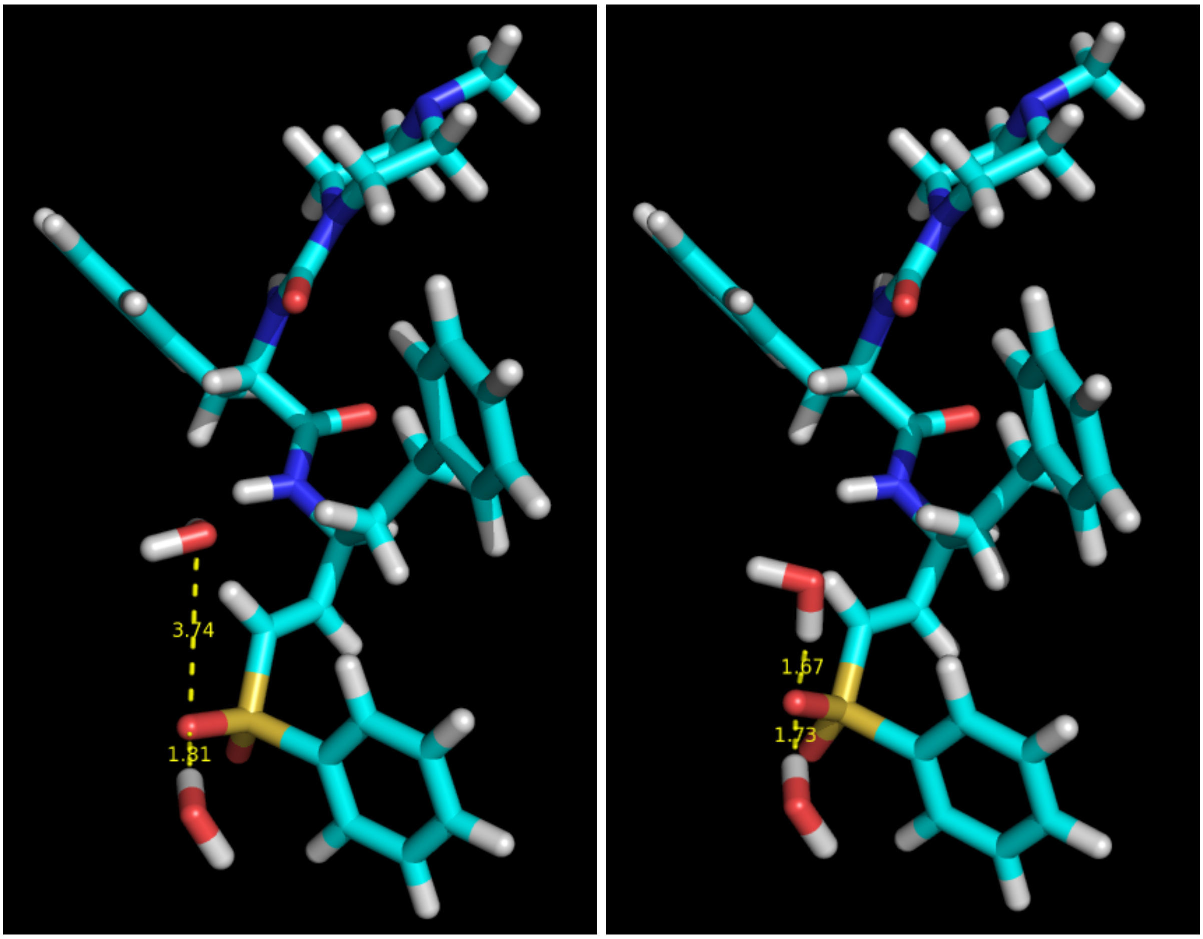


Abbildung 6.17: Reaction paths obtained by 2D scans starting from the non-covalent complex

The energy along the paths is plotted in Figure 6.17. In both scans, the relative energy of the covalent complex decreases in comparison to the singlepoint calculations performed on the original structures, leading to a more exothermic reaction. Thus, the paths become more similar to those starting from the covalent complex (see Figure 6.14). In the computation with QM region *full*, the total energy change is now $-10\frac{\text{kcal}}{\text{mol}}$. With QM region *new*, the total energy change is $-18.4\frac{\text{kcal}}{\text{mol}}$ along the lowest energy path and even $-23.6\frac{\text{kcal}}{\text{mol}}$ if the course of the original path is followed (i. e. the zwitterionic complex is chosen as starting structure). As mentioned before, however, the energies of structures very close to that starting structure might be not reliable, because the starting structure already had an increased N-H distance, which might influence the optimizations where d(N-H) is constraint to a short distance. A distinct feature of the path obtained from the scan with QM system *new* is the energy drop between steps 5 and 6, corresponding to PES points 27_10 and 25_10. It is caused by the movement of a water molecule, which forms an interaction to the sulfone group of the inhibitor (see Figure 6.18). Although this move seems quite random, it was reproducible in different scans with QM region *new*.



(a) Step 5 (27_10)

(b) Step 6 (25_10)

Abbildung 6.18: Interaction of the sulfone group with a water molecule, causing the energy jump between steps 5 and 6 in the path obtained from the 2D scan with QM system *new*

6.3 Umbrella Sampling

Although the *ncov* and *cov* paths approach each other if the QM region for the scan is extended to *full* or *new*, there is still a difference in the total energy change of about $15\frac{\text{kcal}}{\text{mol}}$ depending on the starting point of the scan. For this reason, Umbrella Sampling was performed on the reaction (see section 3.3). This approach differs heavily from the aforementioned scans, as not the internal energies for specific structures are computed, but the relative free energy change along the reaction coordinate (PMF) while averaging over all other dimensions in phase-space. Another advantage of Umbrella Sampling is that only geometries, no energies, are used to compute the PMF. As a consequence, the errors introduced by applying DFTB3 instead of DFT might be smaller, because DFTB3 performs much better in predicting geometries than in calculating energies.[42]

Since the reaction is characterized by the variation of two atomic distances, d(S-C) and d(N-H), these two distances were defined as the reaction coordinates ξ_x and ξ_y in a two-dimensional Umbrella Sampling (see section 5.3.1). In the computation starting from the *cov* structures, the ξ_x coordinate was partitioned into 8 windows ranging from 1.7 to 3.1 Å and ξ_y into 9 windows from 0.9 to 2.5 Å. When starting from the *ncov* structures, d(S-C) ranged from 1.7 to 3.5 Å in steps of 0.2 Å, and d(N-H) from 0.9 to 2.1 Å in steps of 0.1 Å. Only windows in the vicinity of the expected path were sampled. The structures obtained from the original scan performed by Waldemar Waigel were used as initial structures for the MD simulations. Which windows were sampled from which starting structures can be seen in the appendix, Table A.5 and A.6. The force constant for the bias potential was $100\frac{\text{kcal}}{\text{mol}\cdot\text{Å}}$ in both dimensions. The same atoms were fixed as in the original scan. This corresponds to 949 movable atoms in the *cov* simulations and 1021 in the *ncov* MDs. The simulations were performed using the Velocity Verlet integrator with a timestep of 1 fs. Each of them consisted of 10000 equilibration and 20000 production steps. The temperature was kept constant at 300 K with the Nosé-Hoover thermostat. The subtractive QM/MM scheme with electrostatic embedding and option *delM3* was applied as energy interface, where the QM region was treated with DFTB3 and the rest of the system with the OPLSAA force field. The Umbrella Sampling was run with each of the QM regions *small*, *full*, *new* and *full+new*, starting from the *cov* as well as from the *ncov* structures.

After performing the simulations, it was checked if the phase-space along the reaction path is sufficiently covered. This is the case for all eight calculations. The scatterplots are shown in the appendix, Figure A.2 to A.9.

Subsequently, the simulations were processed by WHAM in order to obtain the probability distribution and the PMF. The relative free energies can also be found in the appendix (Table A.7 to A.14). If a value is denoted as “X”, no data is available for the corresponding bin, i. e. the probability for the system to be found there is computed as zero. This happens, because the sampling does not cover the whole phase-space, but simulations were only run on windows in the vicinity of the reaction path.

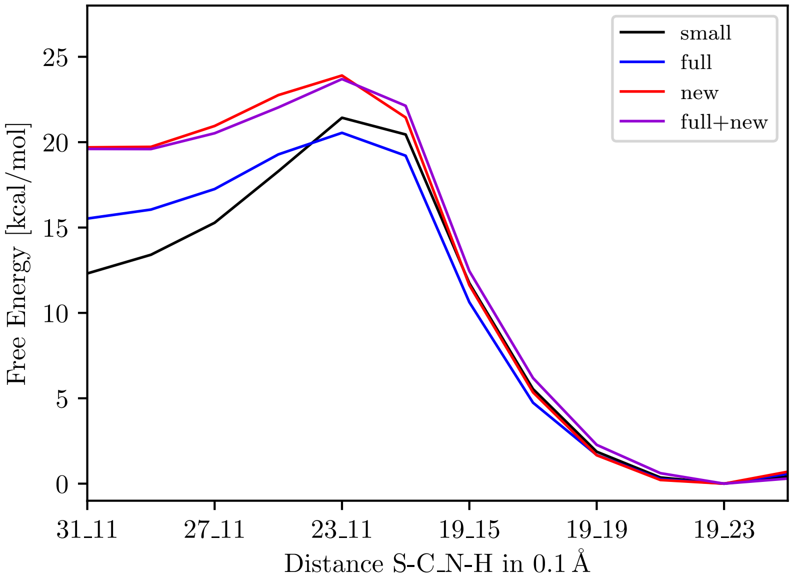


Abbildung 6.19: Reaction paths calculated by Umbrella Sampling, using the structures from the *cov* scan as initial structures for the MD simulations

The free energies along this reaction path (defined by the PES points from the original path) are plotted in Figure 6.19 and 6.20. The PMFs from the *cov* sampling (6.19) all look quite similar. The total change in free energy between educt and product varies between $-12.3\frac{\text{kcal}}{\text{mol}}$ in the calculation with QM region *small* and $-19.7\frac{\text{kcal}}{\text{mol}}$ with QM region *new*. For QM system *full* it is $-15.5\frac{\text{kcal}}{\text{mol}}$. There is nearly no difference in the paths computed with QM regions *new* and *full+new*, confirming that the improvement introduced by the additional atoms is minimal. In contrast to the singlepoint calculations, the transition state at PES point 23_11 is significantly higher in energy than the non-covalent complex, leading to a more reasonable path. The most stable conformation for the covalent complex is at point 19_23 instead of 19_25, which translates to a shorter N-H distance compared to the original scan.

In the cases where the simulations for the Umbrella Sampling are started from the structures obtained by the *ncov* scan, big differences in the PMF depending on the QM region are observed (see Figure 6.20). The path calculated with QM system *small* is

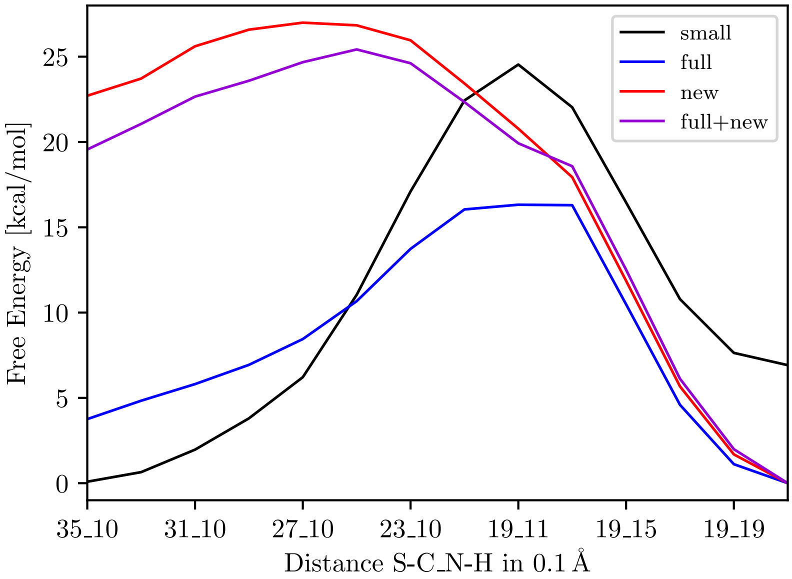


Abbildung 6.20: Reaction paths calculated by Umbrella Sampling, using the structures from the *ncov* scan as initial structures for the MD simulations

endergonic ($\Delta A = 6.8\frac{\text{kcal}}{\text{mol}}$), while all the other paths are exergonic. However, the total change in free energy still differs a lot, especially between the QM systems *full* and *new*. While in the first case it is $-3.8\frac{\text{kcal}}{\text{mol}}$, it decreases to $-22.7\frac{\text{kcal}}{\text{mol}}$ when QM region *new* is applied. *full+new* is again similar to *new* with a total change in free energy of $-19.6\frac{\text{kcal}}{\text{mol}}$. This is identical to the Umbrella Sampling starting from the *cov* structures, where ΔA is also $-19.6\frac{\text{kcal}}{\text{mol}}$ in the computation with QM region *full+new*. The transition state is at PES point 25_10 and has a relative free energy of $25.4\frac{\text{kcal}}{\text{mol}}$, which is also quite close to the one in the Umbrella Sampling starting from the *cov* structures.

These findings suggest that Umbrella Sampling might be the solution to the problem of different energy paths which arise when scans from different ends of the reaction are conducted. If a suitable QM region is chosen (i. e. *new* or *full+new*), the molecular dynamics simulations seem to provide enough flexibility for the system to adjust its conformation during the reaction. However, the question remains why this works only for selected QM regions and not for example with QM region *full*, which contains even more atoms than *new*.

To investigate this question, the computations starting from the *ncov* structures were compared for QM systems *full* and *new*, since they result in very different paths (see 6.20). As only relative free energies are calculated, it cannot be determined from the plot which end of the path induces the difference: Either the free energy for the non-covalent complex is higher when calculated with QM region *new*, or the free energy of the covalent complex is lower. In order to solve this ambiguity, RMSD values were computed for the MD simulations at both ends of the path (PES points 35_10 and 19_21). The root mean square deviation (RMSD) is a means for the comparison of two structures.[130] It is evaluated as

$$RMSD = \sqrt{\frac{1}{N}\sum_{i}^{N} |R_i - R_{i,ref}|^2} \tag{6.1}$$

where N is the number of atoms, and R_i and $R_{i,ref}$ represent the atomic position of atom i in the current structure and in a reference structure.[130] The starting structure of the MD was chosen as the reference, and only atoms that are part of one of the QM regions were included in the calculation. This corresponds to the atoms of QM system *full+new*.

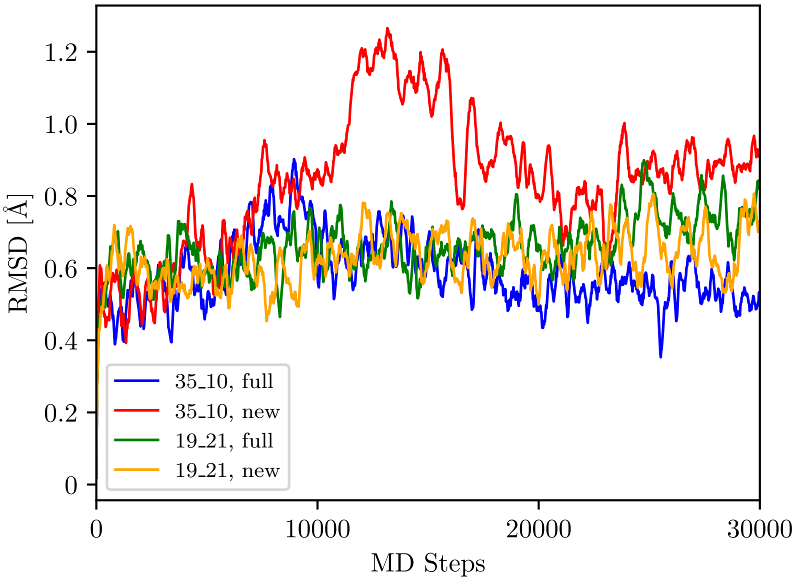
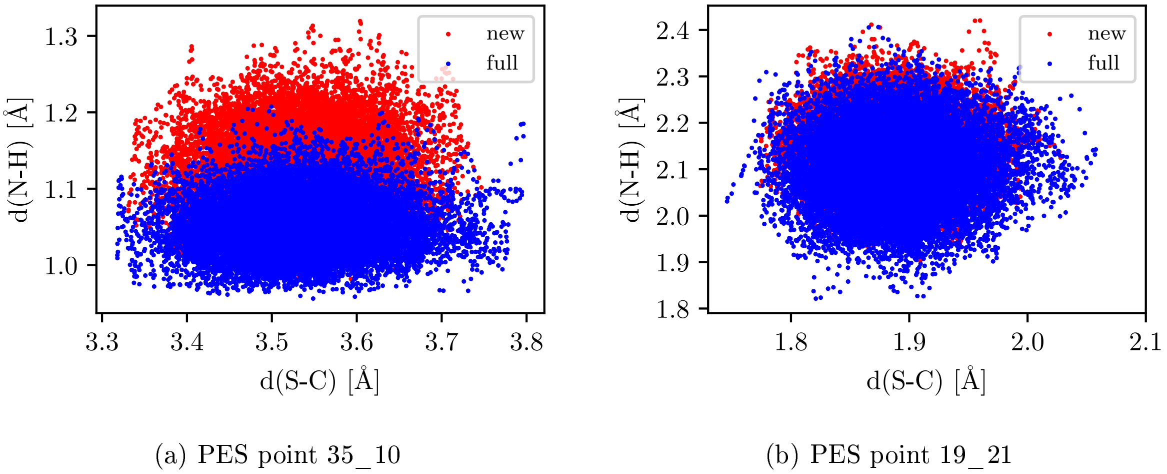


Abbildung 6.21: RMSD values during the MD simulations at both ends of the reaction path in the Umbrella Samplings with QM regions *full* and *new*

The RMSD values of all four simulations are plotted in Figure 6.21. During the simulations at PES point 19_21, the RMSD increases to around 0.8 Å, independently of the choice of the QM region. In the MDs at PES point 35_10, however, the RMSDs during the last two thirds of the simulation, which correspond to the production steps for the Umbrella Sampling, differ significantly. While the RMSD remains relatively low with QM system *full* (in the end it is around 0.5 Å), it rises up to more than 1.2 Å with QM system *new* and ends up at 0.9 Å. A higher RMSD corresponds to stronger movement during the simulation. Stronger movement leads to a higher standard deviation in the probability distribution, which causes WHAM to compute a higher free energy at this point (see also section 3.3.4). As a consequence of the RMSD analysis, it can be stated that the difference between the PMFs obtained with QM regions *full* and *new* originates mainly from differences at the non-covalent end of the reaction path.



(a) PES point 35_10

(b) PES point 19_21

Abbildung 6.22: Scatterplots of the MD simulations at PES points 35_10 and 19_21 with QM systems *full* and *new*

This observation can be confirmed by looking at the distributions of the atomic distances d(S-C) and d(N-H) in the simulations at both ends of the path, which are shown as scatterplots in Figure 6.22. There is nearly no difference in the distribution for the covalent complex (PES point 19_21), whereas the distribution for the non-covalent complex (PES point 35_10) is significantly broader in the simulation with QM system *new*, especially in the N-H direction. Since those two distances are the only piece of information which is transferred to WHAM in order to evaluate the PMF, this directly leads to a higher free energy in comparison to QM region *full*, where the distribution is narrower.

A further explanation of this behavior might be found in the potential energy surfaces computed with the two different QM systems. Despite the insufficient adjustment of the environment during the scans, the results obtained from the calculations starting from the non-covalent complex should describe the surface around this starting point well enough to get some general trends. From Figure 6.15, it can be seen that the non-covalent area of the surface (i. e. at big S-C distances) is quite flat when computing it with QM region *new*, while in the computation with QM system *full* the energy decreases for small N-H distances. Consequently, a system moving on these surfaces during an MD simulation will cover a broader range in N-H distance when QM region *new* is applied, which perfectly agrees with the observations from the scatterplots.

6.4 Molecular Dynamics Simulation

To sum it up, two different ways to compute the (free) energy change during the formation of the rhodesain-K11777 complex have been described in the previous sections:[98]

In the 2D scans (section 6.2), the system is moved along the reaction coordinate by applying a constraint. A local optimization is performed at each step, i. e. there is only a small adjustment in the environment. This can lead to different paths depending on the starting structure, since the system does not reach the global energy minimum at the endpoint of the path (see Figure 6.23, white arrows). This description corresponds to a reaction which is much faster than the adjustment of the environment. First, the reaction takes place, and in a second step, which is not covered by the computation, the rest of the system relaxes to find the optimal conformation.

In Umbrella Sampling (section 6.3), on the other hand, the system is moved along the reaction coordinate by applying a restraint. An MD simulation is performed at each step, allowing a perfect adjustment of the environment if the simulation time is sufficient. As not only one point in phase-space is taken into account, but an average over all other coordinates, the paths starting from the covalent and the non-covalent complex are the same. This is indicated by the red line in Figure 6.23, which marks the global minimum at each step, where the ensemble obtained by the MD has its peak. This description corresponds to a reaction, where the environment adjusts faster than the reaction takes place.

Now the question arises, which of these two approaches is the appropriate one. How fast does the environment really adjust to the reaction?

To answer this question, it would be necessary to simulate the reaction without any constraints or restraints that influence the velocity of atomic movements. Of course, such a simulation without any external bias will never cover the whole reaction, because the system will most probably not cross the activation barrier to go from non-bonded to bonded state. But it is possible to simulate at least a part of the reaction by starting at the transition state. Then the system should relax to the covalent complex, while releasing a reaction energy of around 20 to $25\frac{\text{kcal}}{\text{mol}}$. The distribution of this energy can be observed over time. If the energy is distributed immediately over the whole enzyme, this indicates a very fast adjustment of the environment. If, however, the energy remains in the vicinity of the active site, the movement of the environment seems to be rather slow.

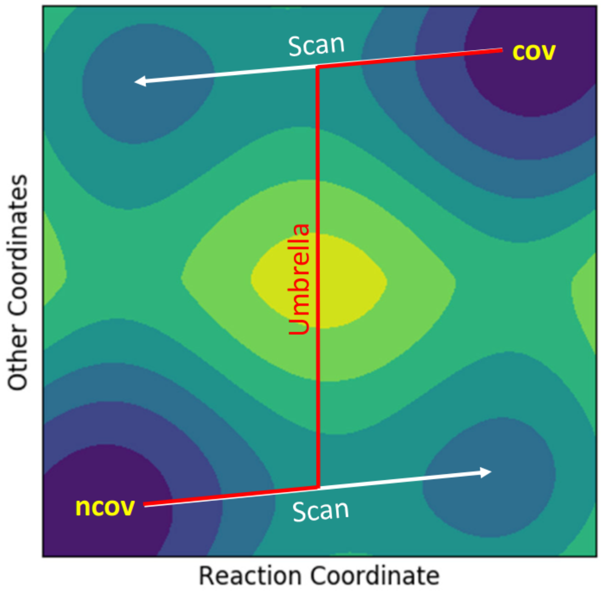


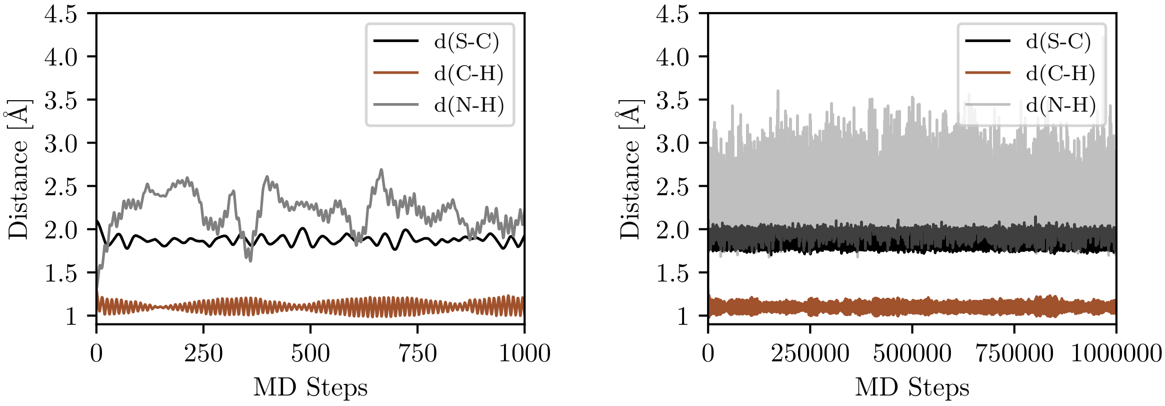
Abbildung 6.23: Reaction paths on a schematic potential energy surface, obtained by either scans (white) or Umbrella Sampling (red)

As the starting structure for the simulation, PES point 21_13 from the *cov* path was chosen. This corresponds to a point next to the transition state, but closer to the covalent end of the path, since the system should be forced to move in this direction (see Figure 6.10). The calculation was performed using the additive QM/MM interface, where QM system *small* (see Figure A.1) was treated with DFTB3 and the rest of the system with the OPLSAA force field. The QM/MM interactions were evaluated using electrostatic embedding with the option *delM3*. As in the scans, 949 atoms around the catalytic dyad were allowed to move.

First the system was equilibrated for 100000 steps with a timestep of 1 fs. During the equilibration, the distances that define the progress of the reaction were fixed by the Rattle algorithm.[131] Besides d(S-C) and d(N-H), which were already used in the scans, the distance d(C-H) between the proton and the carbon atom of the inhibitor, where a bond is formed during the reaction, was also constrained. The molecule was heated from 0 K to 300 K during the first 10000 steps. Subsequently, the temperature was kept constant with the Nosé-Hoover thermostat.[97]

The simulation of the reaction was started from the final structure of the equilibration run. It was performed in NVE mode with a starting temperature of 300 K, i. e. no thermostat was applied. This leads to a conservation of the total energy.

In order to analyze the simulation, the distances d(S-C), d(C-H), and d(N-H) were plotted (see Figure 6.24). The first two decrease from 2.1 to 1.9 Å and from 1.3 to 1.1 Å, respectively. The formation of these bonds is finished almost immediately, after less than 20 steps. As the proton is transferred from the nitrogen of the histidine to the inhibitor carbon, the distance d(N-H) increases from 1.3 Å to around 2.5 Å. Since these atoms are not covalently bound, there are strong fluctuations in their distance. This makes it more difficult to tell at which point the final state is reached, but after 250 fs at the latest, the covalent complex has formed. Such a fast reaction could be expected, because the simulation was started near the transition state. So the system does not have to cross any activation barrier before relaxing to the final covalent state.



(a) first 1000 steps

(b) whole simulation

Abbildung 6.24: Important atomic distances during the MD simulation

In order to observe the energy distribution, the system was partitioned into zones of 3 Å width around the active site. The position of the active site was defined as the geometrical center of Cys-S and Inh-C. The distance of each atom to this active site was computed from the structure after the equilibration. There were 14 atoms in the innermost 3 Å sphere, 86 atoms in the zone between 3 and 6 Å, and 223 atoms in the zone between 6 and 9 Å. Zones with a bigger distance to the active site were not taken into account, since these include atoms that are fixed during the simulation, which influences the kinetic energy.

The average temperature in these zones is calculated at each MD step. For this purpose, the kinetic energy in zone Z is computed as

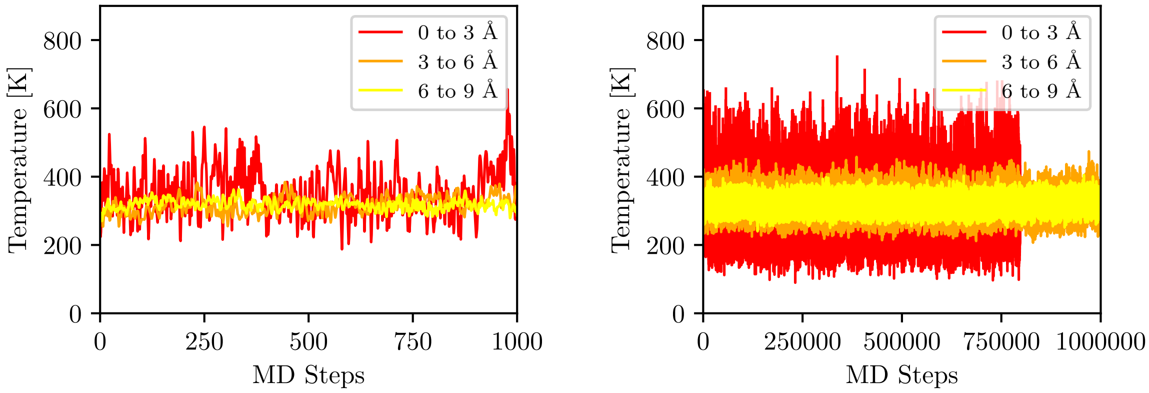
$$E_{kin,Z} = \sum_{i}^{N_Z} \frac{1}{2} m_i v_i^2 \tag{6.2}$$

where N_Z is the number of atoms in zone Z, and m_i and v_i are mass and velocity of atom i respectively. The average temperature can then be evaluated as

$$T_Z = \frac{2 \cdot E_{kin,Z}}{\mathrm{k_B} \cdot N_f} \tag{6.3}$$

with the degrees of freedom $N_f = 3 \cdot N_Z$.[97]

The temperature in the three zones around the active site is depicted in Figure 6.25. No significant accumulation of energy can be seen in any of the zones at any time. The temperature fluctuations seem to be suspiciously strong in the innermost zone, but the comparison to a region consisting of 14 other randomly chosen atoms showed that this is due to the small number of atoms.



(a) first 1000 steps

(b) whole simulation

Abbildung 6.25: Average temperature in zones around the active site

But what would be expected if the energy released from the reaction accumulated in one zone? For a number of atoms N, the kinetic energy can be computed from the temperature T:[20]

$$E_{kin} = N \cdot \frac{3}{2}\mathrm{k_B}T \tag{6.4}$$

By rearranging Equation 6.4, it is possible to compute the temperature increase for the case that the difference in potential energy $\Delta E \approx 20\frac{\mathrm{kcal}}{\mathrm{mol}}$ is completely converted to kinetic energy and distributed over N atoms:

$$T = \frac{2 \cdot \Delta E}{3N \cdot \mathrm{k_B}} \tag{6.5}$$

If all the released energy remained focused in the innermost zone containing 14 atoms, the temperature would increase by around 480 K. As this is not observed during the simulation, the energy must be distributed at least to the next zone, i. e. over 100 atoms. There it would lead to a temperature increase of around 70 K. This is already inside the magnitude of the fluctuations and could thus hardly be recognized in the plots. This problem gets even worse when increasing the region further outside of the 6 Å zone, because as more and more atoms are added the temperature difference decreases. If all 949 movable atoms were taken into account, it would be only 7 K.

Although it is difficult to make a definitive statement about any but the innermost zone, the simulation indicates that the energy released from the reaction distributes immediately over the enzyme. This leads to the assumption that the environment adjusts very fast to the reaction. A description by Umbrella Sampling should thus give a more realistic path than a scan, as it allows the environment to adjust at every reaction step.

7 Summary

Within this work, an additive and a subtractive QM/MM interface were implemented into CAST. The interactions between QM and MM system are described via electrostatic embedding. Link atoms are used to saturate dangling bonds originating from the separation of QM and MM system. Available energy evaluation methods to be combined include force fields (OPLSAA and AMBER), semi-empirical programs (Mopac and DFTB$^+$), and quantum-chemical methods (from Gaussian, Orca, and Psi4). Both the additive and the subtractive interface can deal with periodic boundary conditions. The subtractive scheme was extended to enable QM/QM, three-layer, and multi-center calculations. Another feature only available within the subtractive interface is the microiteration procedure for local optimizations.

The novel QM/MM methods were applied to the investigation of the reaction path for the complex formation between rhodesain and K11777. Benchmark calculations show a very good agreement with results from Gaussian-ONIOM. When comparing the relative energies obtained with different options to a computation where the whole system was treated with the "QM method" DFTB3, the electrostatic embedding scheme with option *delM3* gives the best results. *delM3* means that atoms with up to three bonds distance to the QM region are ignored when creating the external charges. This is done in order to avoid a double counting of Coulomb interactions between QM and MM system. The embedding scheme for the inner system in a three-layer calculation, however, does not have a significant influence on the energies. The same is true for the choice of the coupling scheme: Whether the additive or the subtractive QM/MM interface is applied does not alter the results significantly. The choice of the QM region, though, proved to be an important factor. As can be seen from the comparison of the QM systems *full* and *new*, bigger is not always better here. Instead, one has to make sure not to separate important (polar) interactions by the QM/MM border.

After this benchmark study with singlepoint calculations, the various possibilities of CAST were used to approximate the solution of a remaining problem: The predicted reaction energy for the formation of the rhodesain-K11777 complex differs significantly depending on the starting point of the reaction path. The reason for this is assumed to be an inadequate adjustment of the environment during the scans, which leads to a better stabilization of the starting structure in comparison to the final structure. The first approach to improve this adjustment was performing the relaxed scan with a bigger QM region (*full* or *new*) instead of the minimal QM system *small.* While the paths starting from the covalent complex do not change significantly, those starting from the non-covalent complex become more exothermic, leading to a higher similarity of *cov* and *ncov* paths. Nevertheless, the difference of the reaction energy is still around $15\frac{\text{kcal}}{\text{mol}}$, which is far from a perfect agreement. For this reason, Umbrella Samplings were run. Here, the adjustment of the environment is not done by local optimizations like in the scans, but by MD simulations. This has the advantage that the system can cross barriers and reach different local minima. The relative free energies obtained by Umbrella Samplings with QM regions *new* and *full+new* are nearly identical, independently of the starting point of the calculation. Thus, ΔA evaluated by these computations can be assumed to reproduce the real energy change best. An MD simulation that was started from the transition state in order to mimic a "real-time" reaction indicates a very fast adjustment of the environment during the formation of the complex. This confirms that Umbrella Sampling is probably better suitable to describe the reaction path than a scan, where the environment can never move strong enough to leave the current local minimum.

8 Zusammenfassung

In dieser Arbeit wurden ein additives und ein subtraktives QM/MM-Interface in CAST implementiert. Die Wechselwirkungen zwischen QM- und MM-System werden durch elektrostatische Einbettung beschrieben. Link-Atome dienen dazu, lose Bindungen abzusättigen, die durch die Trennung von QM- und MM-System entstehen. Als Methoden zur Energieberechnung, die kombiniert werden können, stehen Kraftfelder (OPLSAA und AMBER), semiempirische Programme (Mopac und DFTB$^+$) und quantenchemische Verfahren (aus Gaussian, Orca und Psi4) zur Verfügung. Sowohl das additive als auch das subtraktive Interface können mit periodischen Randbedingungen verwendet werden. Erweiterungen des subtraktiven Schemas ermöglichen Berechnungen mit QM/QM, drei Schichten oder mehreren QM-Zentren. Ebenfalls nur im subtraktiven Interface verfügbar ist die lokale Optimierung mittels Mikroiterationsschema.

Die neuen QM/MM-Methoden wurden auf die Untersuchung des Reaktionspfades für die Komplexbildung zwischen Rhodesain und K11777 angewandt. Benchmark-Rechnungen zeigen eine sehr gute Übereinstimmung mit Ergebnissen aus Gaussian-ONIOM. Vergleicht man die relativen Energien, die man mit verschiedenen Optionen erhält, mit einer Rechnung, in der das vollständige System mit der „QM-Methode“ DFTB3 behandelt wird, ergibt das elektrostatische Einbettungsschema mit Option *delM3* die besten Ergebnisse. *delM3* bedeutet, dass Atome mit einem Abstand von bis zu drei Bindungen zur QM-Region bei der Erstellung der externen Ladungen nicht berücksichtigt werden. Dadurch wird eine doppelte Zählung von Coulomb-Wechselwirkungen zwischen QM- und MM-System vermieden. Das Einbettungsschema für das innerste System in einer Drei-Schichten-Rechnung hat jedoch kaum einen Einfluss auf die Energien. Dies gilt auch für die Wahl des Kopplungsschemas: Ob das additive oder das subtraktive Interface verwendet wird, ändert die Ergebnisse nicht signifikant. Im Gegensatz dazu ist die Wahl der QM-Region ein wichtiger Faktor, wie sich herausstellte. Wie man aus dem Vergleich der QM-Systeme *full* und *new* erkennt, bedeutet größer hier nicht immer besser. Stattdessen muss sichergestellt werden, dass keine wichtigen (polaren) Wechselwirkungen durch die QM/MM-Grenze getrennt werden.

Nach dieser Benchmark-Studie mit Einzelpunkt-Rechnungen wurden die vielfältigen Möglichkeiten aus CAST genutzt, um sich der Lösung eines verbleibenden Problems anzunähern: Die Reaktionsenergie, die für die Bildung des Rhodesain-K11777-Komplexes vorhergesagt wird, unterscheidet sich deutlich, je nachdem, welchen Startpunkt man für den Reaktionspfad wählt. Grund dafür ist vermutlich eine unzureichende Anpassung der Umgebung während der Scans, welche zu einer besseren Stabilisierung der Startstruktur im Vergleich zur Endstruktur führt. Als erster Ansatz, diese Anpassung zu verbessern, wurde der Relaxierungs-Scan mit einer größeren QM-Region (*full* oder *new*) anstelle des minimalen QM-Systems *small* durchgeführt. Während sich die vom kovalenten Komplex ausgehenden Pfade kaum ändern, werden diejenigen, die vom nicht-kovalenten Komplex ausgehen, exothermer, wodurch sich *cov*- und *ncov*-Pfade einander annähern. Dennoch beträgt die Differenz der Reaktionsenergie noch immer etwa $15\frac{\text{kcal}}{\text{mol}}$, was weit von einer perfekten Übereinstimmung entfernt ist. Aus diesem Grund wurden Umbrella-Samplings durchgeführt. Dabei passt sich die Umgebung nicht wie in den Scans durch lokale Optimierungen an, sondern durch MD-Simulationen. Das hat den Vorteil, dass das System Barrieren überwinden und verschiedene lokale Minima erreichen kann. Die relativen freien Energien, die man aus Umbrella-Samplings mit den QM-Regionen *new* und *full+new* erhält, sind unabhängig vom Startpunkt der Rechnung nahezu identisch. Daher kann man annehmen, dass die echte Energieänderung durch den ΔA-Wert aus diesen Rechnungen am besten abgebildet wird. Eine MD-Simulation, die ausgehend vom Übergangszustand gestartet wurde, um eine „Echtzeit“-Reaktion nachzuahmen, deutet auf eine sehr schnelle Anpassung der Umgebung während der Komplexbildung hin. Dies bestätigt, dass Umbrella-Sampling wahrscheinlich besser dazu geeignet ist, den Reaktionspfad zu beschreiben, als ein Scan, bei dem sich die Umgebung niemals stark genug bewegen kann, um das aktuelle lokale Minimum zu verlassen.

9 Danksagung

Zuallererst gilt mein Dank Prof. Dr. Bernd Engels für die Ermöglichung dieser Arbeit. Er hatte immer ein offenes Ohr für Fragen und Probleme, gewährte mir aber auch viel Freiheit, um eigenen Ideen nachzugehen. Zudem möchte ich mich bei Prof. Dr. Volker Engel dafür bedanken, dass er sich als Zweitgutachter zur Verfügung gestellt hat.

Des Weiteren bedanke ich mich bei allen Mitgliedern der Arbeitsgruppe für den guten Zusammenhalt und die angenehme Stimmung. Von ihnen konnte man nicht nur jederzeit fachliche Unterstützung erhalten, sondern wir hatten auch viel Spaß zusammen in Kaffeepausen und bei gemeinsamen Unternehmungen. Ein ganz spezieller Dank gebührt hierbei Uschi, der guten Seele des Arbeitskreises, die mir jederzeit bei Fragen (nicht nur) organisatorischer Natur mit Rat und Tat zur Seite stand.

An dieser Stelle möchte ich mich auch bei Dustin Kaiser und Sara Wirsing für das Korrekturlesen dieser Arbeit bedanken.

Zu guter Letzt bedanke ich mich ganz herzlich bei meiner Familie und meinen Freunden, die mich während dieser nicht immer leichten Zeit unterstützt haben. Besonders in der finalen Schreibphase im coronabedingten Home-Office haben sie mich immer wieder motiviert und aufgebaut.

Literatur

1. J.-M. André, “The Nobel Prize in Chemistry 2013: The Alliance of Newton’s Apple and Schrödinger’s Cat”, *Chemistry International* **2014**, *36*, 2–7.
2. A. Warshel, M. Karplus, “Calculation of ground and excited state potential surfaces of conjugated molecules. I. Formulation and parametrization”, *Journal of the American Chemical Society* **1972**, *94*, 5612–5625.
3. A. Warshel, M. Levitt, “Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme”, *Journal of Molecular Biology* **1976**, *103*, 227–249.
4. M. Levitt, S. Lifson, “Refinement of protein conformations using a macromolecular energy minimization procedure”, *Journal of Molecular Biology* **1969**, *46*, 269–279.
5. J. A. Sordo, “Computational contributions to chemistry, biological chemistry and biophysical chemistry: the 2013 Nobel Prize in Chemistry”, *Analytical and Bioanalytical Chemistry* **2014**, *406*, 1825–1828.
6. U. C. Singh, P. A. Kollman, “A combined ab initio quantum mechanical and molecular mechanical method for carrying out simulations on complex molecular systems: Applications to the CH3Cl + Cl- exchange reaction and gas phase protonation of polyethers”, *Journal of Computational Chemistry* **1986**, *7*, 718–730.
7. T. Vreven, K. Morokuma, “Hybrid Methods: ONIOM(QM:MM) and QM/MM” in *Annual Reports in Computational Chemistry, Bd. 2*, Elsevier, **2006**, S. 35–51.
8. D. Bakowies, W. Thiel, “Hybrid Models for Combined Quantum Mechanical and Molecular Mechanical Approaches”, *The Journal of Physical Chemistry* **1996**, *100*, 10580–10594.
9. F. Maseras, K. Morokuma, “IMOMM: A new integrated ab initio + molecular mechanics geometry optimization scheme of equilibrium structures and transition states”, *Journal of Computational Chemistry* **1995**, *16*, 1170–1179.
10. S. Humbel, S. Sieber, K. Morokuma, “The IMOMO method: Integration of different levels of molecular orbital approximations for geometry optimization of large systems: Test for n-butane conformation and SN2 reaction: RCl+Cl-”, *The Journal of Chemical Physics* **1996**, *105*, 1959–1967.

11. M. Svensson, S. Humbel, R. D. J. Froese, T. Matsubara, S. Sieber, K. Morokuma, “ONIOM: A Multilayered Integrated MO + MM Method for Geometry Optimizations and Single Point Energy Predictions. A Test for Diels-Alder Reactions and Pt(P(t-Bu)3)2 + H2 Oxidative Addition”, *The Journal of Physical Chemistry* **1996**, *100*, 19357–19363.

12. S. Dapprich, I. Komáromi, K. S. Byun, K. Morokuma, M. J. Frisch, “A new ONIOM implementation in Gaussian98. Part I. The calculation of energies, gradients, vibrational frequencies and electric field derivatives”, *Journal of Molecular Structure: THEOCHEM* **1999**, *461-462*, 1–21.

13. T. Vreven, K. S. Byun, I. Komáromi, S. Dapprich, J. A. Montgomery, K. Morokuma, M. J. Frisch, “Combining Quantum Mechanics Methods with Molecular Mechanics Methods in ONIOM”, *Journal of Chemical Theory and Computation* **2006**, *2*, 815–826.

14. L. W. Chung, H. Hirao, X. Li, K. Morokuma, “The ONIOM method: its foundation and applications to metalloenzymes and photobiology”, *WIREs Computational Molecular Science* **2012**, *2*, 327–350.

15. R. P. Magalhães, H. S. Fernandes, S. F. Sousa, “Modelling Enzymatic Mechanisms with QM/MM Approaches: Current Status and Future Challenges”, *Israel Journal of Chemistry*, DOI `10.1002/ijch.202000014`.

16. C. S. S. Teixeira, M. J. Ramos, S. F. Sousa, N. M. F. S. A. Cerqueira, “Solving the Catalytic Mechanism of Tryptophan Synthase: an Emergent Drug Target in the Treatment of Tuberculosis”, *ChemCatChem* **2020**, *12*, 227–237.

17. P. Paiva, S. F. Sousa, P. A. Fernandes, M. João Ramos, “Human Fatty Acid Synthase: A Computational Study of the Transfer of the Acyl Moieties from MAT to the ACP Domain”, *ChemCatChem* **2019**, *11*, 3853–3864.

18. P. Ferreira, S. F. Sousa, P. A. Fernandes, M. J. Ramos, “Improving the Catalytic Power of the DszD Enzyme for the Biodesulfurization of Crude Oil and Derivatives”, *Chemistry – A European Journal* **2017**, *23*, 17231–17241.

19. T. Schirmeister, J. Kesselring, S. Jung, T. H. Schneider, A. Weickert, J. Becker, W. Lee, D. Bamberger, P. R. Wich, U. Distler, S. Tenzer, P. Johé, U. A. Hellmich, B. Engels, “Quantum Chemical-Based Protocol for the Rational Design of Covalent Inhibitors”, *Journal of the American Chemical Society* **2016**, *138*, 8332–8335.

20. F. Jensen, *Introduction to Computational Chemistry*, Wiley, Chichester, **1998**.

21. P. Koskinen, V. Mäkinen, "Density-functional tight-binding for beginners", *Computational Materials Science* **2009**, *47*, 237–253.

22. G. Groenhof, "Introduction to QM/MM simulations", *Methods in Molecular Biology* **2013**, *924*, 43–66.

23. G. A. Kaminski, R. A. Friesner, J. Tirado-Rives, W. L. Jorgensen, "Evaluation and Reparametrization of the OPLS-AA Force Field for Proteins via Comparison with Accurate Quantum Chemical Calculations on Peptides", *The Journal of Physical Chemistry B* **2001**, *105*, 6474–6487.

24. Y. Zhang, H. Lin, D. G. Truhlar, "Self-Consistent Polarization of the Boundary in the Redistributed Charge and Dipole Scheme for Combined Quantum-Mechanical and Molecular-Mechanical Calculations", *Journal of Chemical Theory and Computation* **2007**, *3*, 1378–1398.

25. L. Cao, U. Ryde, "On the Difference Between Additive and Subtractive QM/MM Calculations", *Frontiers in Chemistry* **2018**, *6*, 89.

26. S. W. Rick, S. J. Stuart, B. J. Berne, "Dynamical fluctuating charge force fields: Application to liquid water", *The Journal of Chemical Physics* **1994**, *101*, 6141–6156.

27. I. H. Hillier, "Chemical reactivity studied by hybrid QM/MM methods", *Journal of Molecular Structure: THEOCHEM*, Computational Chemistry 1997 **1999**, *463*, 45–52.

28. G. Lamoureux, B. Roux, "Modeling induced polarization with classical Drude oscillators: Theory and molecular dynamics simulation algorithm", *The Journal of Chemical Physics* **2003**, *119*, 3025–3039.

29. C. Czeslik, H. Seemann, R. Winter, *Basiswissen Physikalische Chemie*, 4. Aufl., Vieweg+Teubner Verlag, Wiesbaden, **2010**.

30. N. Jonassen, *Electrostatics*, 2. Aufl., Springer, Norwell, **2002**.

31. B. W. Hopkins, G. S. Tschumper, "A multicentered approach to integrated QM/QM calculations. Applications to multiply hydrogen bonded systems", *Journal of Computational Chemistry* **2003**, *24*, 1563–1568.

32. W. Koch, M. C. Holthausen, *A Chemist's Guide to Density Functional Theory*, 2. Aufl., Wiley-VCH, Weinheim, **2001**.

33. A. F. Oliveira, G. Seifert, T. Heine, H. A. Duarte, "Density-functional based tight-binding: an approximate DFT method", *Journal of the Brazilian Chemical Society* **2009**, *20*, 1193–1205.

34. M. Gaus, Q. Cui, M. Elstner, "DFTB3: Extension of the Self-Consistent-Charge Density-Functional Tight-Binding Method (SCC-DFTB)", *Journal of Chemical Theory and Computation* **2011**, *7*, 931–948.

35. M. Elstner, G. Seifert, "Density functional tight binding", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2014**, *372*, 20120483.

36. M. Elstner, D. Porezag, G. Jungnickel, J. Elsner, M. Haugk, T. Frauenheim, S. Suhai, G. Seifert, "Self-consistent-charge density-functional tight-binding method for simulations of complex materials properties", *Physical Review B* **1998**, *58*, 7260–7268.

37. Yang, H. Yu, D. York, Q. Cui, M. Elstner, "Extension of the Self-Consistent-Charge Density-Functional Tight-Binding Method: Third-Order Expansion of the Density Functional Theory Total Energy and Introduction of a Modified Effective Coulomb Interaction", *The Journal of Physical Chemistry A* **2007**, *111*, 10861–10873.

38. Parameters, The DFTB Website, `http://www.dftb.org/parameters/` (besucht am 18. 02. 2020).

39. S. Kaminski, M. Gaus, M. Elstner, "Improved Electronic Properties from Third-Order SCC-DFTB with Cost Efficient Post-SCF Extensions", *The Journal of Physical Chemistry A* **2012**, *116*, 11927–11937.

40. A. Humeniuk, R. Mitric, "Long-range correction for tight-binding TD-DFT", *The Journal of Chemical Physics* **2015**, *143*, 134120.

41. J. G. Brandenburg, S. Grimme, "Accurate Modeling of Organic Molecular Crystals by Dispersion-Corrected Density Functional Tight Binding (DFTB)", *The Journal of Physical Chemistry Letters* **2014**, *5*, 1785–1789.

42. M. Gaus, X. Lu, M. Elstner, Q. Cui, "Parameterization of DFTB3/3OB for Sulfur and Phosphorus for Chemical and Biological Applications", *Journal of Chemical Theory and Computation* **2014**, *10*, 1518–1537.

43. P. Atkins, J. De Paula, *Physical Chemistry*, 8. Aufl., Oxford University Press, Oxford, **2006**.

44. J. Kästner, “Umbrella sampling”, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, *1*, 932–942.

45. C. Chipot, A. Pohorille, *Free Energy Calculations*, Springer, Berlin/Heidelberg, **2007**.

46. M. Souaille, B. Roux, “Extension to the weighted histogram analysis method: combining umbrella sampling with free energy calculations”, *Computer Physics Communications* **2001**, *135*, 40–57.

47. A. Grossfield, WHAM: The Weighted Histogram Analysis Method, `http://membrane.urmc.rochester.edu/sites/default/files/wham/wham_talk.pdf` (besucht am 25. 02. 2020).

48. J. Becker, *Development and implementation of new simulation possibilities in the CAST program package*, PhD Thesis, Würzburg, **2015**.

49. M. Mills, I. Andricioaei, “An experimentally guided umbrella sampling protocol for biomolecules”, *The Journal of Chemical Physics* **2008**, *129*, 114101.

50. P. S. Hudson, J. K. White, F. L. Kearns, M. Hodoscek, S. Boresch, H. Lee Woodcock, “Efficiently computing pathway free energies: New approaches based on chain-of-replica and Non-Boltzmann Bennett reweighting schemes”, *Biochimica et Biophysica Acta (BBA) - General Subjects* **2015**, *1850*, 944–953.

51. J. J. Ruiz-Pernía, E. Silla, I. Tuñón, S. Martí, V. Moliner, “Hybrid QM/MM Potentials of Mean Force with Interpolated Corrections”, *The Journal of Physical Chemistry B* **2004**, *108*, 8427–8433.

52. L. Schumaker, *Spline Functions: Basic Theory*, 3. Aufl., Cambridge University Press, Cambridge, **2007**.

53. C. Grebner, J. Becker, D. Weber, D. Bellinger, M. Tafipolski, C. Brückner, B. Engels, “CAST: A new program package for the accurate characterization of large and flexible molecular systems”, *Journal of Computational Chemistry* **2014**, *35*, 1801–1807.

54. W. L. Jorgensen, J. Tirado-Rives, “The OPLS Potential Functions for Proteins. Energy Minimizations for Crystals of Cyclic Peptides and Crambin”, *Journal of the American Chemical Society* **1988**, *110*, 1657–1666.

55. A. D. MacKerell Jr., D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, M. Karplus, “All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins”, *The Journal of Physical Chemistry B* **1998**, *102*, 3586–3616.

56. Gaussian.com | Expanding the limits of computational chemistry, `https://gaussian.com/` (besucht am 28. 02. 2020).

57. R. M. Parrish, L. A. Burns, D. G. A. Smith, A. C. Simmonett, A. E. DePrince, E. G. Hohenstein, U. Bozkaya, A. Y. Sokolov, R. Di Remigio, R. M. Richard, J. F. Gonthier, A. M. James, H. R. McAlexander, A. Kumar, M. Saitow, X. Wang, B. P. Pritchard, P. Verma, H. F. Schaefer, K. Patkowski, R. A. King, E. F. Valeev, F. A. Evangelista, J. M. Turney, T. D. Crawford, C. D. Sherrill, “Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability”, *Journal of Chemical Theory and Computation* **2017**, *13*, 3185–3197.

58. J. J. P. Stewart, MOPAC 2016, **2016**, `http://openmopac.net/` (besucht am 28. 02. 2020).

59. S. Wirsing, “MULTI SCALE METHODS - The implementation of a QM/MM energy interface class within CAST”, Bachelor Thesis, Würzburg, **2016**.

60. U. Eichler, C. M. Kölmel, J. Sauer, “Combining ab initio techniques with analytical potential functions for structure predictions of large systems: Method and application to crystalline silica polymorphs”, *Journal of Computational Chemistry* **1997**, *18*, 463–477.

61. M. Swart, “AddRemove: A new link model for use in QM/MM studies”, *International Journal of Quantum Chemistry* **2003**, *91*, 177–183.

62. F. Neese, “Software update: the ORCA program system, version 4.0”, *WIREs Computational Molecular Science* **2018**, *8*, e1327.

63. B. Aradi, B. Hourahine, T. Frauenheim, “DFTB+, a Sparse Matrix-Based Implementation of the DFTB Method”, *The Journal of Physical Chemistry A* **2007**, *111*, 5678–5684.

64. DFTB+ USER MANUAL, `http://www.dftbplus.org/fileadmin/DFTBPLUS/public/dftbplus/latest/manual.pdf` (besucht am 04. 03. 2020).

65. F. Neese, F. Wennmohs, W. Schneider, *ORCA - An ab initio, DFT and semiempirical SCF-MO package - Version 4.1.0*, Mülheim a. d. Ruhr, **2018**.

66. T. Okamoto, K. Yamada, Y. Koyano, T. Asada, N. Koga, M. Nagaoka, "A minimal implementation of the AMBER-GAUSSIAN interface for ab initio QM/MM-MD simulation", *Journal of Computational Chemistry* **2011**, *32*, 932–942.

67. PSI4: Open-Source Quantum Chemistry (Manual), `http://www.psicode.org/psi4manual/master/index.html` (besucht am 04. 03. 2020).

68. QMMM - MOPAC Manual, `http://openmopac.net/Manual/QMMM.html` (besucht am 04. 03. 2020).

69. B. W. Hopkins, G. S. Tschumper, "Multicentred QM/QM methods for overlapping model systems", *Molecular Physics* **2005**, *103*, 309–315.

70. O. M. Becker, A. D. MacKerell Jr., B. Roux, M. Watanabe, *Computational Biochemistry and Biophysics*, Marcel Dekker Inc, New York, **2001**.

71. List of quantum chemistry and solid-state physics software, Wikipedia, **2020**, `https://en.wikipedia.org/w/index.php?title=List_of_quantum_chemistry_and_solid-state_physics_software&oldid=945214968` (besucht am 13. 03. 2020).

72. P. J. Steinbach, B. R. Brooks, "New spherical-cutoff methods for long-range forces in macromolecular simulation", *Journal of Computational Chemistry* **1994**, *15*, 667–683.

73. F. Neese, "The ORCA program system", *WIREs Computational Molecular Science* **2012**, *2*, 73–78.

74. std::system - cppreference.com, `https://en.cppreference.com/w/cpp/utility/program/system` (besucht am 23. 03. 2020).

75. Creating Processes - Win32 apps, Library Catalog: docs.microsoft.com, `https://docs.microsoft.com/en-us/windows/win32/procthread/creating-processes` (besucht am 23. 03. 2020).

76. S. Grimme, S. Ehrlich, L. Goerigk, "Effect of the damping function in dispersion corrected density functional theory", *Journal of Computational Chemistry* **2011**, *32*, 1456–1465.

77. T. A. Niehaus, F. D. Sala, "Range separated functionals in the density functional based tight-binding method: Formalism", *Physica Status Solidi B* **2012**, *249*, 237–244.

78. J. W. Ponder, Tinker User's Guide, **2019**, `https://dasher.wustl.edu/tinker/downloads/guide.pdf` (besucht am 25. 03. 2020).

79. XYZ (format) - Open Babel, `http://openbabel.org/wiki/XYZ_%28format%29` (besucht am 25. 03. 2020).

80. M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, G. R. Hutchison, "Avogadro: an advanced semantic chemical editor, visualization, and analysis platform", *Journal of Cheminformatics* **2012**, *4*, 17.

81. Jmol: an open-source Java viewer for chemical structures in 3D, `http://www.jmol.org` (besucht am 25. 03. 2020).

82. W. Humphrey, A. Dalke, K. Schulten, "VMD: visual molecular dynamics", *Journal of Molecular Graphics* **1996**, *14*, 33–38.

83. L.-P. Wang, C. Song, "Geometry optimization made simple with translation and rotation coordinates", *The Journal of Chemical Physics* **2016**, *144*, 214108.

84. G. Schaftenaar, J. Noordik, "Molden: a pre- and post-processing program for molecular and electronic structures", *Journal of Computer-Aided Molecular Design* **2000**, *14*, 123–134.

85. J. A. Rackers, Z. Wang, C. Lu, M. L. Laury, L. Lagardère, M. J. Schnieders, J.-P. Piquemal, P. Ren, J. W. Ponder, "Tinker 8: Software Tools for Molecular Design", *Journal of Chemical Theory and Computation* **2018**, *14*, 5273–5289.

86. V. Zoete, M. A. Cuendet, A. Grosdidier, O. Michielin, "SwissParam: A fast force field generation tool for small organic molecules", *Journal of Computational Chemistry* **2011**, *32*, 2359–2368.

87. Tinker Molecular Modelling - Parameters, `https://dasher.wustl.edu/tinker/distribution/params/` (besucht am 25. 03. 2020).

88. J. M. Berg, L. Stryer, J. L. Tymoczko, *Biochemistry*, 5. Aufl., W.H.Freeman & Co Ltd, New York, **2002**.

89. Amber Workshop - Tutorial A1 - Section 1: Do some editing of the PDB file, `http://ambermd.org/tutorials/advanced/tutorial1_orig/section1.htm` (besucht am 02. 04. 2020).

90. wwPDB: File Format Documentation, `http://www.wwpdb.org/documentation/file-format` (besucht am 02. 04. 2020).

91. RCSB PDB: Homepage, `https://www.rcsb.org/` (besucht am 02. 04. 2020).

92. Pymol - The PyMOL Molecular Graphics System, Version 2.0 Schrödinger, LLC. `https://pymol.org/2/` (besucht am 02. 04. 2020).

93. J. Drenth, *Principles of Protein X-Ray Crystallography*, 3. Aufl., Springer, New York/London, **2010**.

94. PyMOLWiki, `https://pymolwiki.org/index.php/Main_Page` (besucht am 06. 04. 2020).

95. A. Grossfield, WHAM: the weighted histogram analysis method, version 2.0.9, `http://membrane.urmc.rochester.edu/?page_id=126` (besucht am 06. 04. 2020).

96. S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, P. A. Kollman, "Multidimensional free-energy calculations using the weighted histogram analysis method", *Journal of Computational Chemistry* **1995**, *16*, 1339–1350.

97. D. Frenkel, B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 2. Aufl., Academic Press, San Diego, **2001**.

98. E. Rosta, H. L. Woodcock, B. R. Brooks, G. Hummer, "Artificial reaction coordinate "tunneling" in free-energy calculations: The catalytic reaction of RNase H", *Journal of Computational Chemistry* **2009**, *30*, 1634–1641.

99. H. Zheng, S. Wang, Y. Zhang, "Increasing the Time Step with Mass Scaling in Born-Oppenheimer ab initio QM/MM Molecular Dynamics Simulations", *Journal of computational chemistry* **2009**, *30*, 2706–2711.

100. D. Weber, "THE SOLVADD PROGRAM - Conformational sampling of hydration shells based on the formation of hydrogen bonds", Diploma Thesis, Würzburg, **2011**.

101. GaussView 6 | Gaussian.com, `http://gaussian.com/gaussview6/` (besucht am 07. 04. 2020).

102. A. P. Bento, M. Solà, F. M. Bickelhaupt, "Ab initio and DFT benchmark study for nucleophilic substitution at carbon (SN2@C) and silicon (SN2@Si)", *Journal of Computational Chemistry* **2005**, *26*, 1497–1504.

103. S. Bochkanov, ALGLIB - C++/C# numerical analysis library, `https://www.alglib.net/` (besucht am 08. 04. 2020).

104. Calc | LibreOffice - Deutschsprachiges Projekt - Freie Office Suite, `https://de.libreoffice.org/discover/calc/` (besucht am 08. 04. 2020).

105. A. Blondel, M. Karplus, “New formulation for derivatives of torsion angles and improper torsion angles in molecular mechanics: Elimination of singularities”, *Journal of Computational Chemistry* **1996**, *17*, 1132–1141.

106. Bicubic spline interpolation/fitting - ALGLIB, C++ and C# library, `https://www.alglib.net/interpolation/bicubic-spline-interpolation-fitting.php#header2` (besucht am 14. 04. 2020).

107. C. Grebner, *New Tabu-Search Algorithms for the Exploration of Energy Landscapes of Molecular Systems*, PhD Thesis, Würzburg, **2012**.

108. T. Vreven, K. Morokuma, Ö. Farkas, H. B. Schlegel, M. J. Frisch, “Geometry optimization with QM/MM, ONIOM, and other combined methods. I. Microiterations and constraints”, *Journal of Computational Chemistry* **2003**, *24*, 760–769.

109. J. Kästner, S. Thiel, H. Senn, P. Sherwood, W. Thiel, “Exploiting QM/MM Capabilities in Geometry Optimization: A Microiterative Approach Using Electrostatic Embedding”, *Journal of Chemical Theory and Computation* **2007**, *3*, 1064–1072.

110. H. Hu, Z. Lu, J. M. Parks, S. K. Burger, W. Yang, “Quantum mechanics/molecular mechanics minimum free-energy path for accurate reaction energetics in solution and enzymes: Sequential sampling and optimization on the potential of mean force surface”, *The Journal of Chemical Physics* **2008**, *128*, 034105.

111. R. B. Murphy, D. M. Philipp, R. A. Friesner, “A mixed quantum mechanics/molecular mechanics (QM/MM) method for large-scale modeling of chemistry in protein environments”, *Journal of Computational Chemistry* **2000**, *21*, 1442–1457.

112. J. Erdmannsdörfer, “Development and Implementation of Methods Determining Reaction Pathways”, Master Thesis, Würzburg, **2017**.

113. S. Metz, J. Kästner, A. A. Sokol, T. W. Keal, P. Sherwood, “ChemShell—a modular software package for QM/MM simulations”, *WIREs Computational Molecular Science* **2014**, *4*, 101–110.

114. J. C. Meza, R. A. Oliva, P. D. Hough, P. J. Williams, “OPT++: An object-oriented toolkit for nonlinear optimization”, *ACM Transactions on Mathematical Software* **2007**, *33*, 12–es.

115. J. C. Meza, P. D. Hough, P. J. Williams, R. A. Oliva, OPT++: An Object-Oriented Nonlinear Optimization Library, `https://software.sandia.gov/opt++/opt++2.4_doc/html/index.html` (besucht am 05. 05. 2020).

116. R. Ettari, L. Tamborini, I. C. Angelo, N. Micale, A. Pinto, C. De Micheli, P. Conti, “Inhibition of Rhodesain as a Novel Therapeutic Modality for Human African Trypanosomiasis”, *Journal of Medicinal Chemistry* **2013**, *56*, 5637–5658.

117. WHO, Trypanosomiasis, human African (sleeping sickness), `https://www.who.int/news-room/fact-sheets/detail/trypanosomiasis-human-african-(sleeping-sickness)` (besucht am 12. 06. 2020).

118. D. Steverding, D. W. Sexton, X. Wang, S. S. Gehrke, G. K. Wagner, C. R. Caffrey, “Trypanosoma brucei: Chemical evidence that cathepsin L is essential for survival and a relevant drug target”, *International Journal for Parasitology* **2012**, *42*, 481–488.

119. M.-H. Abdulla, T. O’Brien, Z. B. Mackey, M. Sajid, D. J. Grab, J. H. McKerrow, “RNA Interference of Trypanosoma brucei Cathepsin B and L Affects Disease Progression in a Mouse Model”, *PLoS Neglected Tropical Diseases* **2008**, *2*, e298.

120. I. D. Kerr, J. H. Lee, C. J. Farady, R. Marion, M. Rickert, M. Sajid, K. C. Pandey, C. R. Caffrey, J. Legac, E. Hansell, J. H. McKerrow, C. S. Craik, P. J. Rosenthal, L. S. Brinen, “Vinyl Sulfones as Antiparasitic Agents and a Structural Basis for Drug Design”, *The Journal of Biological Chemistry* **2009**, *284*, 25697–25703.

121. T. Schirmeister, A. Welker, “Erfolgreiches Konzept: Proteasen als Zielstrukturen für Antiinfektiva. Neue Angriffspunkte in der Infektionstherapie”, *Pharmazie in unserer Zeit* **2009**, *38*, 564–574.

122. S. S. Santos, R. V. de Araújo, J. Giarolla, O. E. Seoud, E. I. Ferreira, “Searching for drugs for Chagas disease, leishmaniasis and schistosomiasis: a review”, *International Journal of Antimicrobial Agents* **2020**, *55*, 105906.

123. Y. Zhou, P. Vedantham, K. Lu, J. Agudelo, R. Carrion, J. W. Nunneley, D. Barnard, S. Pöhlmann, J. H. McKerrow, A. R. Renslo, G. Simmons, “Protease inhibitors targeting coronavirus and filovirus entry”, *Antiviral Research* **2015**, *116*, 76–84.

124. S.-S. Jean, P.-I. Lee, P.-R. Hsueh, “Treatment options for COVID-19: The reality and challenges”, *Journal of Microbiology, Immunology and Infection* **2020**.

125. W. Waigel, T. Schirmeister, B. Engels, “Investigations about the Inhibition Mechanisms of Fluorinated Vinylsulfones”, unpublished work, **2020**.

126. P. Tao, H. B. Schlegel, “A toolkit to assist ONIOM calculations”, *Journal of Computational Chemistry* **2010**, *31*, 2363–2369.

127. J. Meyr, "QM/QM/MM-Studien - Auswirkungen eines zusätzlichen QM-Bereichs auf die Inhibition von Rhodesain mit K11777", Bachelor Thesis, Würzburg, **2018**.

128. J. Weiser, "Inhibition von Rhodesain - Ansätze zur Beschreibung enzymatischer Hemmung mittels QM/QM/MM-Methoden", Bachelor Thesis, Würzburg, **2019**.

129. M. Asbach, "Inhibition von Cysteinproteasen - Anwendung des QM/QM/MM-Ansatzes am Beispiel der Inhibition von Rhodesain mit K11777", Bachelor Thesis, Würzburg, **2019**.

130. R. Merkl, S. Waack, *Bioinformatik Interaktiv*, 2. Aufl., Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, **2009**.

131. H. C. Andersen, "Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations", *Journal of Computational Physics* **1983**, *52*, 24–34.

A Appendix

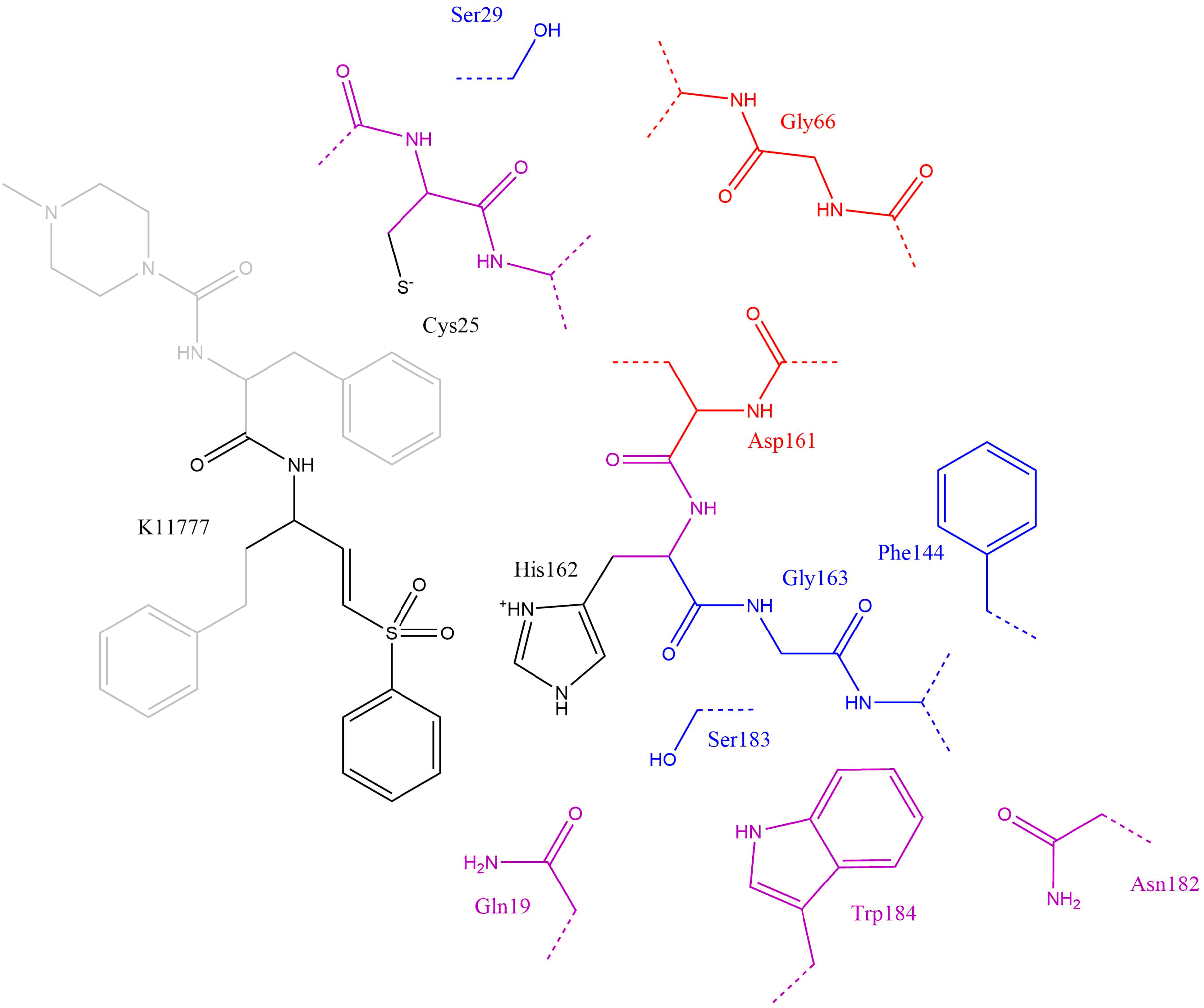


Abbildung A.1: Schematic Illustration of QM Regions *small* (black), *full* (black, violet and blue) and *new* (black, violet and red)

Energy [kcal/mol]	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	d(S-C) [Å]
2.2	17.5	67.4	60	50.9	40.1	28.5	16.9	6.6	0.0	
2.1	24.8	66.7	59.4	50.4	39.8	28.2	16.8	6.6	0.1	
2.0	71.8	66	58.8	49.9	39.5	28.2	16.9	6.8	0.4	
1.9	71.1	65.5	58.3	49.6	39.4	28.3	17.3	7.3	1.0	
1.8	70.7	65.1	58.1	49.6	39.7	28.9	18.1	8.4	2.3	
1.7	70.4	65.0	58.2	49.9	40.4	30	19.6	10.1	4.2	
1.6	69.6	64.7	58.3	50.4	41.4	31.5	21.6	12.5	6.9	
1.5	68.5	63.6	57.8	50.5	42.0	32.9	23.7	15.1	9.8	
1.4	66.1	61.5	55.9	49.6	42.0	33.8	25.5	17.6	12.7	
1.3	61.6	57.6	52.8	47.1	40.8	33.8	26.7	19.8	15.4	
1.2	27.1	25.9	25.1	24.5	38.5	33.2	27.6	21.8	18.1	
1.1	25.2	25.8	25.0	24.4	23.5	23.0	23.3	20.8	18.6	
1.0	29.4	28.0	26.9	24.9	23.0	22.0	21.6	19.9	18.3	
d(N-H) [Å]										

Tabelle A.1: Results of 2D scan starting from covalent complex, calculated with QM region *full*. The lowest energy path is marked in blue.

Energy [kcal/mol]	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	d(S-C) [Å]
2.2	71.3	66.2	59.3	50.4	39.9	28.2	16.7	6.5	0.0	
2.1	70.9	65.7	58.7	49.9	39.4	27.9	16.6	6.4	-0.1	
2.0	70.6	65.3	58.4	49.6	39.2	27.9	16.6	6.6	0.2	
1.9	70.6	65.2	58.2	49.7	39.4	28.3	17.1	7.2	0.9	
1.8	70.7	65.4	58.6	50.1	40.0	29.1	18.2	8.4	2.3	
1.7	71.3	66.0	59.3	51.0	41.3	30.6	20.0	10.5	4.5	
1.6	71.8	66.7	60.3	52.4	43.0	32.7	22.5	13.3	7.5	
1.5	71.9	67.0	60.9	53.5	44.6	35.0	25.3	16.5	11.0	
1.4	70.6	66.1	60.6	53.9	45.8	36.9	27.9	19.6	14.5	
1.3	67.4	63.5	58.9	53.1	46.0	38.2	30.2	22.6	17.9	
1.2	41.5	41.0	40.0	51.1	45.4	39.0	32.2	25.6	21.3	
1.1	32.6	32.4	31.8	30.7	29.7	29.5	29.0	25.4	22.4	
1.0	36.5	35.9	34.8	33.0	31.0	29.3	27.7	24.6	22.1	
d(N-H) [Å]										

Tabelle A.2: Results of 2D scan starting from covalent complex, calculated with QM region *new*. The lowest energy path is marked in red.

Energy [kcal/mol]	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	d(S-C) [Å]
2.2	9.5	9.4	8.3	8.8	10.4	15.3	24.2	-4.5	-10.4	
2.1	9.1	9.1	7.8	8.4	9.9	14.8	23.5	-4.5	-10.2	
2.0	8.6	8.8	7.4	7.9	9.4	14.2	22.8	-4.2	-9.8	
1.9	8.2	8.1	6.9	7.5	9.0	13.6	22.1	-3.7	-9.1	
1.8	7.8	7.7	6.6	7.2	8.6	13.2	21.4	-2.6	-7.8	
1.7	7.6	7.5	6.5	7.1	8.5	12.9	20.8	-0.9	-5.9	
1.6	7.4	7.3	6.5	7.0	8.3	12.6	20.2	1.4	-3.2	
1.5	6.8	6.8	6.0	6.6	7.8	11.9	19.1	4.0	-0.3	
1.4	5.5	5.5	4.9	5.4	6.4	10.3	17.1	6.3	2.6	
1.3	4.0	3.4	2.8	4.1	4.2	7.8	19.2	8.4	5.2	
1.2	1.9	1.1	0.7	1.7	2.9	5.9	12.2	12.8	10.0	
1.1	0.5	-0.1	-0.4	-0.8	0.0	1.3	5.1	7.5	7.2	
1.0	0.0	-0.3	-0.9	-1.5	-1.1	0.1	3.5	6.4	7.1	
d(N-H) [Å]										

Tabelle A.3: Results of 2D scan starting from non-covalent complex, calculated with QM region *full*. The lowest energy path is marked in blue.

Energy [kcal/mol]	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	d(S-C) [Å]
2.2	-3.6	-3.1	-2.4	-2.3	-0.2	-2.1	6.2	-18.2	-23.6	
2.1	-4.1	-3.5	-2.9	-2.8	-0.7	-2.7	5.6	-18.3	-23.6	
2.0	-4.6	-4.0	-3.3	-3.3	-1.2	-3.2	4.9	-18.3	-23.5	
1.9	-5.0	-4.3	-3.7	-3.6	-1.6	-3.6	4.3	-17.9	-23.0	
1.8	-5.2	-4.4	-3.8	-3.8	-1.8	-3.9	3.9	-16.9	-21.9	
1.7	-5.0	-4.1	-3.5	-3.5	-1.6	-3.8	3.6	-15.2	-20.0	
1.6	-4.6	-3.7	-2.6	-2.5	-1.1	-3.1	3.5	-12.8	-17.4	
1.5	-4.1	-3.4	-1.9	-2.3	-0.3	-3.2	3.0	-10.1	-14.3	
1.4	-2.8	-3.7	-2.0	-2.6	-0.8	-4.0	1.9	-7.3	-11.2	
1.3	-3.5	-4.5	-2.3	-2.4	-1.9	-5.5	0.8	-5.0	-8.1	
1.2	-4.2	-5.1	-3.3	-3.4	-1.8	-7.2	-0.1	-0.3	-5.1	
1.1	-3.6	-3.3	-2.1	-2.2	-1.6	-7.3	-4.1	-4.6	-5.5	
1.0	0.0	-0.4	0.2	-2.2	-2.1	-8.0	-5.3	-4.4	-5.7	
d(N-H) [Å]										

Tabelle A.4: Results of 2D scan starting from non-covalent complex, calculated with QM region *new*. The lowest energy path is marked in red.

	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.1							**19_21**	19_21	
1.9							**19_19**	19_19	
1.7							**19_17**	19_17	
1.5						19_15	**19_15**	19_15	
1.3					21_13	**21_13**	21_13	19_15	
1.1	**31_11**	**29_11**	**27_11**	**25_11**	**23_11**	23_11	21_13	21_13	
0.9	31_11	29_11	27_11	25_11	23_11	23_11	23_11	21_13	
d(N-H) [Å]									

Tabelle A.5: Windows sampled for the *cov* Umbrella Sampling, each with the PES point of the respective initial structure. The original path is marked bold.

	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.1									**19_21**	19_21	
2.0									19_21	19_21	
1.9									**19_19**	19_19	
1.8									19_19	19_19	
1.7									**19_17**	19_17	
1.6									19_17	19_17	
1.5									**19_15**	19_15	
1.4									19_15	19_15	
1.3									**19_13**	19_13	
1.2									19_13	19_13	
1.1								21_10	**19_11**	19_11	
1.0	**35_10**	**33_10**	**31_10**	**29_10**	**27_10**	**25_10**	**23_10**	**21_10**	19_11	19_11	
0.9	35_10	33_10	31_10	29_10	27_10	25_10	23_10	21_10	19_11	19_11	
d(N-H) [Å]											

Tabelle A.6: Windows sampled for the *ncov* Umbrella Sampling, each with the PES point of the respective initial structure. The original path is marked bold.

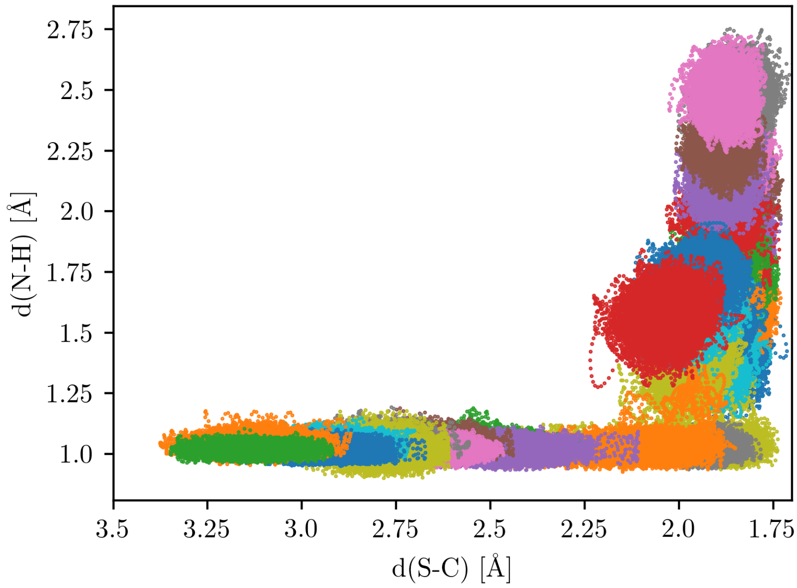


Abbildung A.2: Coverage of the reaction path in the Umbrella Sampling with QM region *small*, using the structures from the *cov* scan as initial structures for the MD simulations

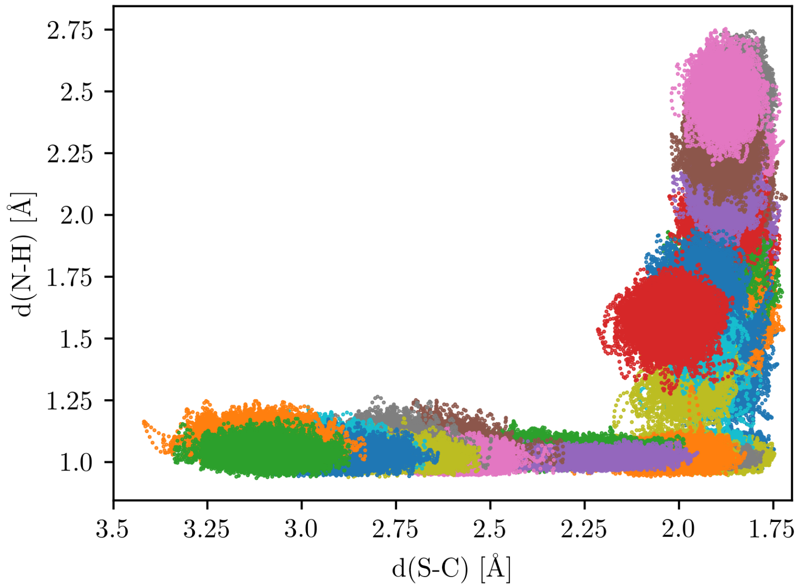


Abbildung A.3: Coverage of the reaction path in the Umbrella Sampling with QM region *full*, using the structures from the *cov* scan as initial structures for the MD simulations

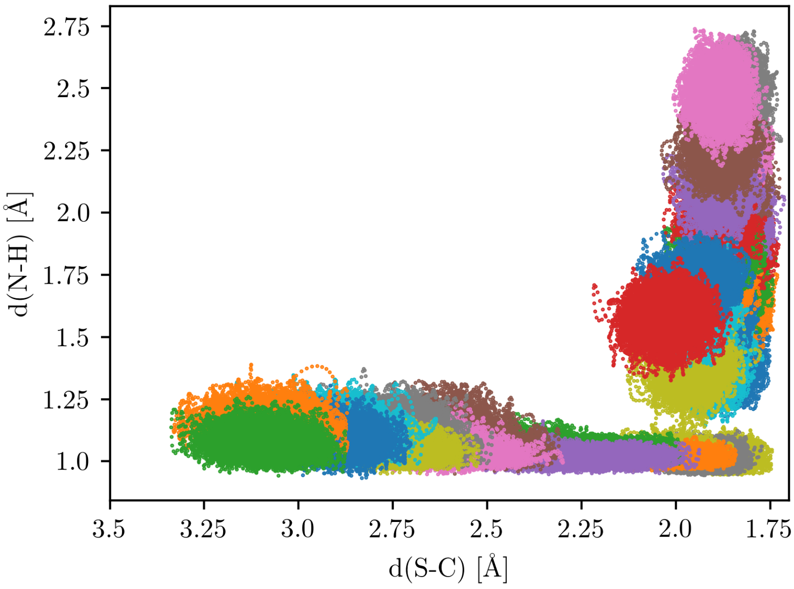


Abbildung A.4: Coverage of the reaction path in the Umbrella Sampling with QM region *new*, using the structures from the *cov* scan as initial structures for the MD simulations

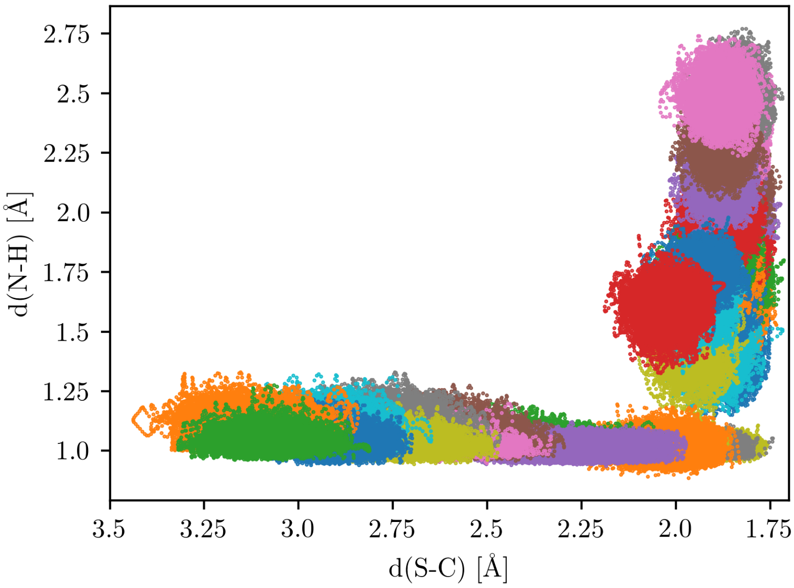


Abbildung A.5: Coverage of the reaction path in the Umbrella Sampling with QM region *full+new*, using the structures from the *cov* scan as initial structures for the MD simulations

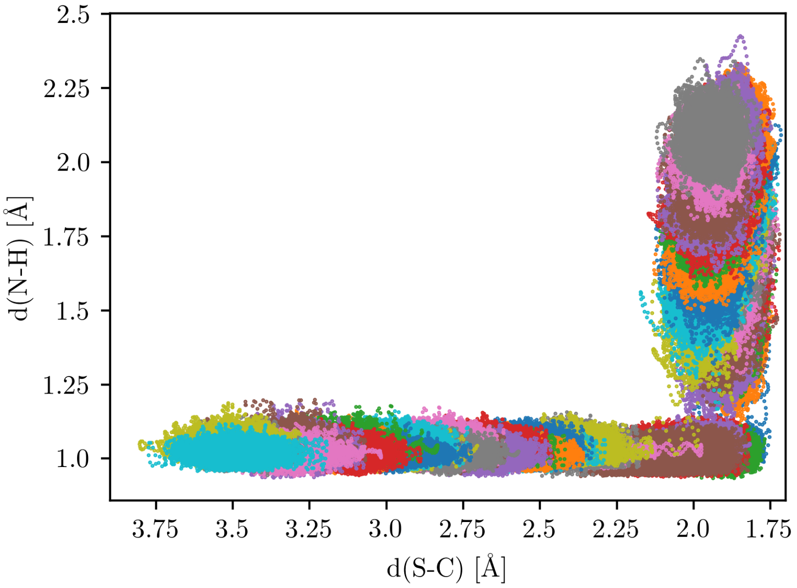


Abbildung A.6: Coverage of the reaction path in the Umbrella Sampling with QM region *small*, using the structures from the *ncov* scan as initial structures for the MD simulations

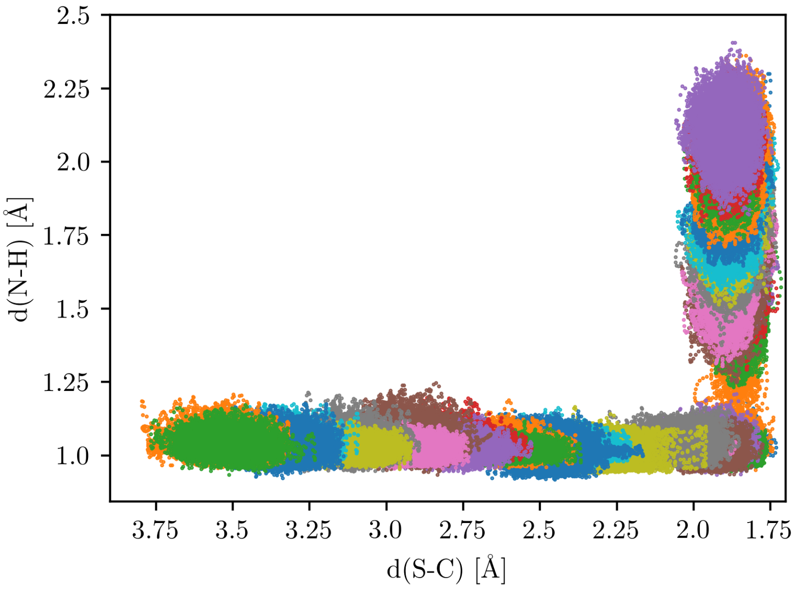


Abbildung A.7: Coverage of the reaction path in the Umbrella Sampling with QM region *full*, using the structures from the *ncov* scan as initial structures for the MD simulations

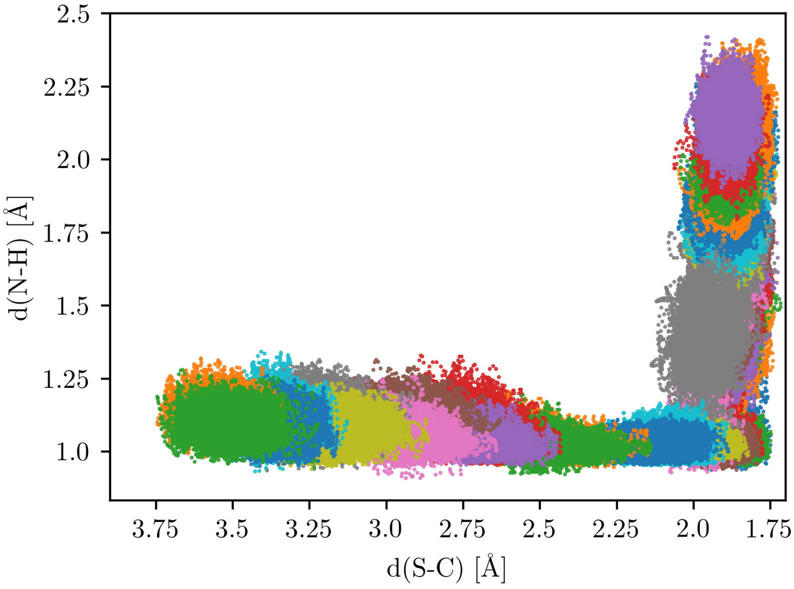


Abbildung A.8: Coverage of the reaction path in the Umbrella Sampling with QM region *new*, using the structures from the *ncov* scan as initial structures for the MD simulations

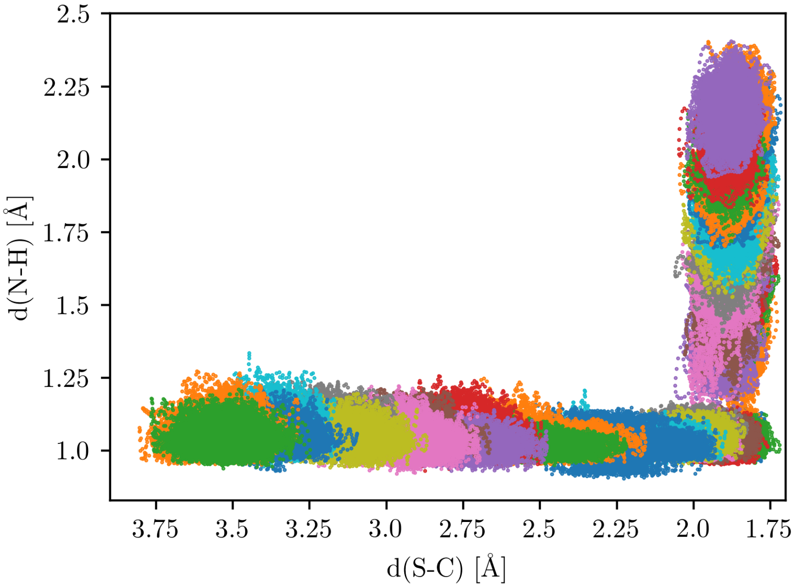


Abbildung A.9: Coverage of the reaction path in the Umbrella Sampling with QM region *full+new*, using the structures from the *ncov* scan as initial structures for the MD simulations

PMF [kcal/mol]	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.5	X	X	X	X	X	2.6	0.4	3.4	
2.3	X	X	X	X	X	2.4	0.0	3.0	
2.1	X	X	X	X	X	2.2	0.4	3.5	
1.9	X	X	X	X	X	4.1	1.9	5.1	
1.7	X	X	X	X	15.8	10.2	5.5	8.8	
1.5	X	X	X	X	21.6	16.1	11.7	15.0	
1.3	X	X	X	X	24.5	20.4	18.0	21.6	
1.1	12.3	13.4	15.3	18.3	21.4	22.7	22.2	26.2	
0.9	15.0	15.8	17.8	20.8	23.8	25.5	25.2	29.5	
d(N-H) [Å]									

Tabelle A.7: Relative free energies obtained by Umbrella Sampling with QM region *small*, using the structures from the *cov* scan as initial structures for the MD simulations

PMF [kcal/mol]	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.5	X	X	X	X	X	3.0	0.6	3.7	
2.3	X	X	X	X	X	2.0	0.0	3.2	
2.1	X	X	X	X	X	2.1	0.3	3.5	
1.9	X	X	X	X	X	3.5	1.7	4.9	
1.7	X	X	X	X	X	9.9	4.7	7.9	
1.5	X	X	X	X	21.7	15.8	10.6	13.9	
1.3	16.5	18.0	18.6	22.2	X	19.2	16.2	19.7	
1.1	15.5	16.1	17.3	19.3	20.5	19.1	17.2	21.2	
0.9	18.7	19.0	19.9	21.9	23.3	21.8	20.0	24.1	
d(N-H) [Å]									

Tabelle A.8: Relative free energies obtained by Umbrella Sampling with QM region *full*, using the structures from the *cov* scan as initial structures for the MD simulations

PMF [kcal/mol]	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.5	X	X	X	X	X	3.2	0.7	3.9	
2.3	X	X	X	X	X	1.8	0.0	3.2	
2.1	X	X	X	X	X	1.8	0.2	3.4	
1.9	X	X	X	X	X	4.1	1.7	4.8	
1.7	X	X	X	X	16.7	10.7	5.3	8.6	
1.5	X	X	X	X	23.6	16.6	11.6	15.1	
1.3	18.6	18.8	20.3	22.8	X	21.4	17.9	21.5	
1.1	19.7	19.7	20.9	22.8	23.9	22.7	21.0	24.9	
0.9	23.7	23.8	24.6	26.1	26.6	25.4	23.7	27.6	
d(N-H) [Å]									

Tabelle A.9: Relative free energies obtained by Umbrella Sampling with QM region *new*, using the structures from the *cov* scan as initial structures for the MD simulations

PMF [kcal/mol]	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.5	X	X	X	X	X	1.7	0.3	3.3	
2.3	X	X	X	X	X	2.4	0.0	3.2	
2.1	X	X	X	X	X	2.6	0.6	3.7	
1.9	X	X	X	X	X	4.7	2.3	5.4	
1.7	X	X	X	X	X	11.5	6.2	9.4	
1.5	X	X	X	X	X	17.6	12.4	15.8	
1.3	19.2	19.3	20.4	23.2	X	22.1	18.6	22.3	
1.1	19.6	19.6	20.5	22.0	23.7	22.5	21.1	25.2	
0.9	23.1	22.9	23.5	24.9	26.0	24.8	23.6	27.9	
d(N-H) [Å]									

Tabelle A.10: Relative free energies obtained by Umbrella Sampling with QM region *full+new*, using the structures from the *cov* scan as initial structures for the MD simulations

PMF [kcal/mol]	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.1	X	X	X	X	X	X	X	9.6	6.2	9.5	
2.0	X	X	X	X	X	X	X	9.7	6.4	9.7	
1.9	X	X	X	X	X	X	X	10.2	7.0	10.3	
1.8	X	X	X	X	X	X	X	11.2	8.1	11.5	
1.7	X	X	X	X	X	X	X	13.1	10.0	13.5	
1.6	X	X	X	X	X	X	X	15.6	12.7	16.2	
1.5	X	X	X	X	X	X	X	18.1	15.7	19.3	
1.4	X	X	X	X	X	X	X	20.3	18.6	22.3	
1.3	X	X	X	X	X	X	X	22.0	21.4	25.3	
1.2	3.4	3.6	5.6	7.7	10.8	16.8	23.7	23.7	24.2	28.4	
1.1	0.3	0.9	2.2	4.1	6.6	11.7	17.9	21.8	24.0	28.9	
1.0	0.0	0.6	1.9	3.7	6.1	10.9	16.9	22.0	24.2	29.1	
0.9	5.0	4.9	7.1	9.7	10.8	16.6	20.6	25.2	28.2	X	
d(N-H) [Å]											

Tabelle A.11: Relative free energies obtained by Umbrella Sampling with QM region *small*, using the structures from the *ncov* scan as initial structures for the MD simulations

PMF [kcal/mol]	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.1	X	X	X	X	X	X	X	1.0	0.0	3.3	
2.0	X	X	X	X	X	X	X	1.9	0.3	3.6	
1.9	X	X	X	X	X	X	X	2.9	1.1	4.3	
1.8	X	X	X	X	X	X	X	4.0	2.5	5.8	
1.7	X	X	X	X	X	X	X	6.2	4.6	7.9	
1.6	X	X	X	X	X	X	X	8.7	7.4	10.7	
1.5	X	X	X	X	X	X	X	11.7	10.5	13.9	
1.4	X	X	X	X	X	X	X	15.5	13.6	17.0	
1.3	X	X	X	X	X	X	X	18.5	16.3	19.8	
1.2	4.1	5.7	6.2	7.0	9.8	13.3	18.1	19.7	18.3	21.5	
1.1	3.0	4.2	5.4	6.5	8.3	10.6	14.3	16.2	16.3	20.5	
1.0	3.8	4.8	5.8	6.9	8.4	10.7	13.7	16.0	16.3	20.5	
0.9	8.7	9.0	10.8	12.1	13.0	13.6	16.8	19.8	20.5	X	
d(N-H) [Å]											

Tabelle A.12: Relative free energies obtained by Umbrella Sampling with QM region *full*, using the structures from the *ncov* scan as initial structures for the MD simulations

PMF [kcal/mol]	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.1	X	X	X	X	X	X	X	2.1	0.0	3.2	
2.0	X	X	X	X	X	X	X	2.4	0.7	4.0	
1.9	X	X	X	X	X	X	X	3.0	1.7	5.0	
1.8	X	X	X	X	X	X	X	4.5	3.4	6.7	
1.7	X	X	X	X	X	X	X	6.8	5.7	9.1	
1.6	X	X	X	X	X	X	X	11.2	8.7	12.2	
1.5	X	X	X	X	X	X	X	14.9	11.9	15.4	
1.4	X	X	X	X	X	X	X	18.1	15.0	18.6	
1.3	19.2	20.4	22.9	24.3	25.1	X	X	20.9	17.9	21.6	
1.2	19.5	20.9	23.2	24.5	25.7	27.8	X	23.7	20.8	24.6	
1.1	20.6	21.9	24.0	25.2	26.1	26.7	26.1	23.0	20.8	24.7	
1.0	22.7	23.7	25.6	26.6	27.0	26.8	26.0	23.4	21.3	25.2	
0.9	27.9	28.2	29.9	30.7	32.0	30.5	30.5	28.6	25.8	X	
d(N-H) [Å]											

Tabelle A.13: Relative free energies obtained by Umbrella Sampling with QM region *new*, using the structures from the *ncov* scan as initial structures for the MD simulations

PMF [kcal/mol]	3.5	3.3	3.1	2.9	2.7	2.5	2.3	2.1	1.9	1.7	d(S-C) [Å]
2.1	X	X	X	X	X	X	X	1.6	0.0	3.3	
2.0	X	X	X	X	X	X	X	2.4	0.8	4.1	
1.9	X	X	X	X	X	X	X	3.5	2.0	5.2	
1.8	X	X	X	X	X	X	X	5.4	3.7	7.0	
1.7	X	X	X	X	X	X	X	7.8	6.1	9.4	
1.6	X	X	X	X	X	X	X	11.8	9.1	12.5	
1.5	X	X	X	X	X	X	X	16.9	12.5	16.0	
1.4	X	X	X	X	X	X	X	19.8	15.7	19.3	
1.3	18.0	19.8	X	24.0	X	X	X	22.4	18.6	22.0	
1.2	18.3	19.6	21.6	22.9	24.3	26.1	26.1	24.3	21.3	25.3	
1.1	18.4	19.8	21.6	22.8	24.1	25.2	24.4	21.9	19.9	23.9	
1.0	19.6	21.1	22.7	23.6	24.7	25.4	24.6	22.3	20.5	24.3	
0.9	24.8	25.4	27.0	27.5	28.6	29.5	27.0	25.7	26.5	X	
d(N-H) [Å]											

Tabelle A.14: Relative free energies obtained by Umbrella Sampling with QM region *full+new*, using the structures from the *ncov* scan as initial structures for the MD simulations