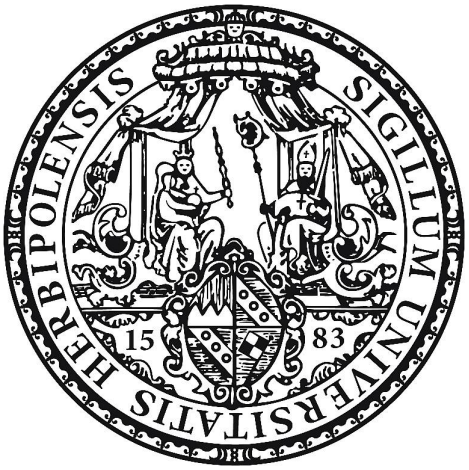
Development and Implementation of new Simulation Possibilities in the CAST program package



Dissertation zur Erlangung des

naturwissenschaftlichen Doktorgrades

der Julius-Maximilians-Universität Würzburg

vorgelegt von

Johannes Becker

aus Würzburg

Würzburg 2015

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: _____

List of Abbreviations

AA	All Atom
ABF	Adaptive Biasing Force
AM1	Austin Model 1
AMBER	Assisted Model Building with Energy Refinement
AMOEBA	Atomic Multipole Optimized Energetics for Biomolecular Simulation
APBS	Adaptive Poisson Boltzmann Solver
BAR	Bennett Acceptance Ratio
B3LYP	Becke 3 Parameters Lee-Yang-Parr
BH	Basin Hopping
BLYP	Becke Lee-Yang-Parr
CAST	Conformational Analysis and Seach Tool
CFF	Consistent Force Field
CHARMM	Chemistry at Harvard Macromolecular Mechanics
COSMO	Conductor Like Screening Model
DFT	Density Functional Theory
FEP	Free Energy Perturbation
FFTW	Fast Fourier Transform of the West
GAFF	Generalized Amber Force Field
GB	Generalized Born
GBSA	Generalized Born Surface Area
GPU	Graphics Processing Unit
HF	Hartree Fock
IUPAC	International Union of Pure and Applied Chemistry
LC	Linked Cell
MC	Monte Carlo
MCMM	Markov Chain Monte Carlo
MD	Molecular Dynamics
MIC	Minimum Image Criterion
MM	Molecular Mechanics
MOPAC	Molecular Orbital Package
MP2	Møller Plesset Perturbation Theory Second Order
MPI	Message Parsing Interface
NUMA	Non Uniform Memory Access
OpenMP	Open Multiprocessor
OPLS-AA	Optimized Potential for Liquid Simulations - All Atom

PBC	Periodic Boundary Conditions
PBI	Perylene Bisimide
PCM	Polarized Continuum Model
PES	Potential Energy Surface
PM3	Parametric Model 3
PM6	Parametric Model 6
PM6-DH2	Parametric Model 6 with Dispersion and Hydrogen Bond Correction
PMF	Potential of Mean Force
QM/MM	Quantum Mechanics / Molecular Mechanics
SAPT	Symmetry Adapted Perturbation Theory
SASA	Solvent Accessible Surface Area
SMP	Symmetric Multiprocessor
SOS	Simple Overlap Sampling
SPME	Smooth Particle Mesh Ewald
TI	Thermodynamic Integration
TS	Tabu Search
UA	United Atom
UMA	Uniform Memory Access
US	Umbrella Sampling
VDW	Van der Waals
VMD	Visual Molecular Dynamics
WHAM	Weighted Histogram Analysis Method
ZDO	Zero Differential Overlap

Contents

1 A brief history about simulation methods **1**

1.1 Monte Carlo . . . 2

1.2 Molecular dynamics . . . 4

1.2.1 Langevin dynamics . . . 6

1.2.2 Brownian dynamics . . . 6

1.2.3 Newtonian molecular dynamics . . . 7

1.3 Sampling in different ensembles . . . 10

1.3.1 Canonical Ensemble (NVT) . . . 11

1.3.2 Isothermal-Isobaric ensemble (NPT) . . . 13

1.4 Free energy calculations . . . 14

2 Aim of the work **19**

3 Theory **21**

3.1 Overview . . . 21

3.2 Force fields . . . 22

3.3 Free energy . . . 30

3.3.1 Mathematical formulation . . . 30

3.3.2 Ergodicity principle . . . 34

3.4 Calculating free energy differences . . . 35

3.4.1 Umbrella sampling . . . 36

3.4.2 Free energy perturbation . . . 39

4 Implementation **41**

4.1 Performance . . . 41

4.1.1 Linked cell algorithm . . . 41

4.1.2 Parallelization . . . 44

4.2 Smooth Particle Mesh Ewald method . . . 48

4.3 Molecular Dynamics . . . 55

4.3.1 General features . . . 55

4.3.2 Integrators . . . 56
4.3.3 Temperature and Pressure . . . 58
4.3.4 Constraints . . . 64
4.3.5 Boundary conditions . . . 66
4.4 Free energy algorithms . . . 69
4.4.1 Umbrella sampling . . . 69
4.4.2 Free energy perturbation . . . 71
4.5 PDB converter (CHARMM parameters) . . . 79
4.6 MPI interface to TeraChem . . . 82
4.7 Implicit solvation: Generalized Born (GB) . . . 85

5 Application **91**
5.1 Preliminary calculations . . . 91
5.1.1 Parallel performance . . . 91
5.1.2 Linked Cell performance . . . 94
5.1.3 Free energy algorithms . . . 96
5.2 Global optimization using CAST/TeraChem . . . 104
5.3 Flexibility of PBI sidechains . . . 107
5.4 Relative binding free energies . . . 114
5.4.1 Technical details about the simulations . . . 116
5.4.2 Results and discussion . . . 118

6 Summary **135**

7 Zusammenfassung **137**

Bibliography **139**

A Appendix **157**
A.1 CAST Results . . . 157
A.1.1 Vacuum . . . 157
A.1.2 Solvent . . . 157

1. A brief history about simulation methods

The Nobel Prize in chemistry 2013 was awarded jointly to Martin Karplus, Michael Levitt and Arieh Warshel *for the development of multiscale models for complex chemical systems.*

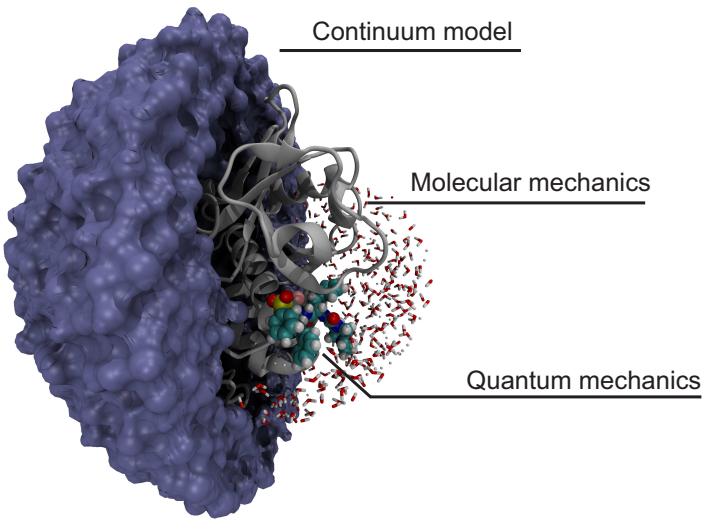


Figure 1.1.: Example of a large multiscale system. The inner most and smallest part can be treated by quantum chemical methods. The surroundings can be treated by atomistic molecular mechanics methods or, depending on the size and computational power, continuum methods.

In the following chapter an overview about the historical development of simulation techniques is given. Since this is a very large field some parts are only mentioned briefly. The main focus will be on the techniques which are relevant for the presented work. For the remaining methods the reader is referred to the respective literature [1–6] for more detailed information.

Theoretical and computational methods have become increasingly important in the

global field of chemistry and especially biochemistry. This is underlined by the award of the 2013 Nobel Prize in chemistry to three of the pioneers for the development of computational methods to describe and analyze complex chemical systems and reactions. In this regard, one has to distinguish between the underlying method to calculate for example energy and gradients of the system and methodologies to efficiently sample the phase space or configurations of the system. The models for the description of these systems can generally be broken down into quantum chemical [7–11], semi-empirical [12–14] and classical models [15]. In the quantum chemical model the description is broken down to the smallest particles of interest, the electrons and nuclei of the atoms. The development of these quantum chemical methods was already started over 75 years ago and many important contributions were made by well known scientists like Planck, Bohr, de Broglie, Heisenberg and Schrödinger. Techniques to also model the surroundings of a system are nowadays mostly based on the work of Coulomb and on van-der-Waals interactions. Development in this field was driven by the work of Westheimer [16–18] and Allinger [19–21] which used these empirical potentials to minimize the structure of systems. Allinger was also the first one to develop computer code for these empirical potentials [22–24], and therefore provided one of the first molecular mechanics programs which he called MM1, MM2,... etc. A first application was done by Scheraga and Gibson [25–27] which used simplified versions of these potentials to perform statistical mechanics simulations for the energy minimization of proteins. Inter- and intramolecular potentials derived by quantum chemical calculations were pioneered by Lifson and Warshel [28–31] with the development of the Consistent Force Field (CFF) method (for a detailed description of force fields see Section 3.2). Building upon these developments, especially in the force field department, it was now possible to apply these methods in combination with sampling algorithms like Monte Carlo (see Section 1.1) and molecular dynamics (Section 1.2).

1.1. Monte Carlo

Originating in the examination of neutron diffusion, it was the idea of Neumann and Ulam to use a probabilistic analogue for a definite mathematical problem and solve it by a stochastic sampling experiment. These sampling experiments were based on the generation of random numbers and a set of finite, normally few, mathematical operations of arithmetic and logical nature. Metropolis further refined the method with the help of the first computer (MANIAC) in Los Alamos. The Metropolis Monte Carlo (MC) [32] is able to generate a trajectory in phase space, and sample a given statistical

ensemble. It belongs to the class of Markov chain Monte Carlo (MCMM) [33] methods which are widely used in statistics, economics, physics and computing science [34], and was also used for the development of the simulated annealing [35] technique. The MC method is used to construct a random walk where the construction follows the condition that the visit of a particular position $\mathbf{r}^N$ is proportional to the Boltzmann factor $\exp\left[-\beta U(\mathbf{r}^N)\right]$. Here N denotes the number of particles, U the potential energy and $\beta = 1/(k_B T)$ with T as the temperature and k_B as the Boltzmann constant. The scheme proposed by Metropolis looks like follows:

1. select a particle at random and calculate its energy $U(\mathbf{r}^N)$

2. give the particle a random displacement $r^{'} = r + \triangle$ and calculate its new energy $U(\mathbf{r}^{'N})$

3. accept the move $\mathbf{r}^N$ to $\mathbf{r}^{'N}$ with the probability (Metropolis criterion)

$$p_A = \min\left(1, \exp\left(-\frac{\triangle E}{kT}\right)\right) \tag{1.1}$$

   Here E is the energy, T the temperature and k the Boltzmann constant.

4. if $\triangle E \leq 0$ the new solution is accepted

5. if $\triangle E > 0$ select a random number between 0 and 1: if p_A is smaller than the random number the new move is accepted

In case the new solution is rejected the old solution is kept as start for the next step. Variations and generalizations of this algorithm were later proposed by Barker [36], Hastings [33] and Peskun [37]. Especially Hastings made significant improvements. The Metropolis-Hastings algorithm can generate states with any arbitrary probability density distribution. The only precondition is that the density can be calculated at any point which is visited during the simulation. Instead of the energy difference a probability is used as the criterion.

$$p_A = \min\left(1, \frac{W(y)P(x_i|y)}{W(x), P(y|x_i)}\right) \tag{1.2}$$

Here W denotes the acceptance distribution, the conditional probability to accept the proposed state x' and P the proposal distribution which is the conditional probability

of proposing a state y given x_i. If the density p_A is constant around the current position in a discrete interval and otherwise zero, the Metropolis algorithm is recovered.
Further generalizations were proposed by Smith [38] and Schmeiser [39]. Despite these improvements, in most practical Monte Carlo simulations different configurations of a given system are created using the Metropolis algorithm. Based on these configurations the averages and expectation values of physical parameters, e.g pressure, density etc., can be calculated. For this, numerous iterations of the algorithm are performed until the system reaches thermal equilibrium which means the probability of the single configurations equal the Boltzmann distribution. In thermal equilibrium the different configurations are generated with the probability

$$W(x) = \frac{1}{Z} e^{-\beta E(x)} \tag{1.3}$$

Here E is the energy, $\beta = 1/(k_B T)$ and Z is the partition function. The final value is obtained by averaging over the single values. It can also be used as a stochastic optimization algorithm, for example to find a global minimum on a multidimensional hyperplane. In this case the original algorithm has to be modified. A famous example of such an optimization algorithm based on Monte Carlo is the basin hopping which was first proposed by Scheraga [40, 41] and later improved by Wales [42].

1.2. Molecular dynamics

Another powerful tool for detailed microscopic modeling on an atomistic scale is molecular dynamics (MD) [5, 43]. It is a widely used tool in chemistry, physics and material science. It provides a scheme for the study of the natural time evolution of a given system and allows the prediction of static and dynamic properties of substances directly from the underlying interactions between its atoms and molecules. In contrast to static calculations these dynamic simulations monitor time-dependent processes in the molecular systems. This can be done by numerically solving an equation of motion, based on which the motions are performed by the molecule. Hence, it provides information about time dependence and magnitude of fluctuations in both positions and velocities. This is in sharp contrast to the Monte Carlo technique which is normally only based on positions. The equations of motions used for the simulation can be, depending on the level of accuracy, based on classical equations (Newton), stochastic equations (Langevin) or Brownian equations.
A crucial aspect of these simulations is to know the timescale in which the desired observed motion resides. Macromolecules in general exert a huge range of characteristic

motions on a large difference of timescales. These motions can vary from very fast and localized motions, like the stretching vibration of a hydrogen bond, to very slow and large scale motions, like the folding of a whole protein. An overview of the different motions and their timescale can be found in Table 1.1.

Type	**Example**	**Timescale**
Local		
Atomic fluctuations	Stretching vibrations	Femtoseconds (fs) to picoseconds (ps)
Side chain motion		
Medium-scale motions		
Loop motion	Active site conformation adaption, binding specificity	Nanoseconds (ns) to microseconds (μs)
Terminal-arm motion		
Rigid-body motion		
Large-scale motions		
Domain motion	Hinge bending motion, allosteric transitions	Mircroseconds (μs) to milliseconds (ms)
Subunit motion		
Global motions		
Helix-coil transition	Hormone activation, protein functionality	Milliseconds (ms) to hours (h)
Folding/unfolding		
Subunit association		

Table 1.1.: Overview of the different timescales and types of motions in macromolecules. The range extends from local motions, which happen in femto- to picosenconds up to global motions in the rang of milliseconds to even hours. Table has been reproduced and modified from ref [6] with permission from Taylor & Francis.

One of the most important aspects to note here is the fact that many of those different motions are coupled to each other. A large scale dynamic motion, or transition, always includes medium scale motions in the process. In the same way the medium-scaled motions are almost always dependent on local motions, for example vibrations. This has pretty severe implications on a dynamic simulation, since the study of large scale motions (which are normally the ones of biological relevance) has to also include the fast-scale motions of the system. These fast scale motions are thus the limiting factor when it comes to length and time step of the calculation.

1.2.1. Langevin dynamics

Sometimes the interesting part of a system doesn't involve the complete system. For example one may be interested in the motion of only one part of a protein, or the effects of different solvents on the motion. If one has to include all specific motions and properties by using an all-atom model with explicit solvent molecules the computation can easily become very expensive. One solution is to eliminate all explicit solvent degrees of freedom and handle them in a more general way leading to implicit solvent models. Another way is to refer to stochastic dynamics instead of molecular dynamics. For the stochastic dynamics the underlying equation is the Langevin equation [44, 45]

$$m_i \ddot{r}_i = -\nabla_i U(r) - m_i \beta_i v_i(t) + R_i(t) \tag{1.4}$$

The first term $\nabla_i U(r)$ on the right side accounts for the interactions and is equal to the term used in newtonian molecular dynamics. The other two terms are defining for the Langevin equation and are used for modeling solvent effects. $m_i \beta_i v_i(t)$ is a force which arises from the friction of the solvent. It is proportional to the velocity v_i of the particle and a friction coefficient β_i which is related to the diffusion constant D_i. The third term R_i is a random force and models the collisions between solute and solvent on a stochastic basis. With this equation, energy is extracted from the system due to the friction, and is added to the system by the use of the stochastic forces. The random force is obtained by a Gaussian distribution with zero mean and variance:

$$\begin{aligned} \langle R_i(t) \rangle &= 0 \\ \langle R_i(t) R_i(0) \rangle &= 6 m_i k_B T \triangle(t) = 2 D_i \triangle(t) \end{aligned} \tag{1.5}$$

1.2.2. Brownian dynamics

The high viscosity of the solvent can play an important role on the motions of the solute. If the motions of the solute include large displacements of the molecular surface, for example if heavy atoms are moving around in aqueous solution, these movements can get damped heavily owing to the before mentioned viscosity. If these damping effects get big enough the internal forces become negligible and the motion has the nature of a random walk. In a different system these internal forces may be not of interest at all. If that is the case a simplification for the equations of motion can be introduced which leads to the Brownian equation of motion. This is

$$v_i(t) = \dot{r}_i = \frac{-\nabla V_i(r) + F_i^{mean} + R_i(t)}{m_i \beta_i} \tag{1.6}$$

Here R_i denotes a stochastic force, F_i^{mean} is a mean force which resembles an average force based on degrees of freedom which are not explicitly sampled during the simulation, V_i is the potential, m_i the mas and β_i the friction coefficient. The Brownian equation can simply be derived form the Langevin equation. The term on left hand side of the Langevin equation may be neglected if it is small compared to the right hand side force term. Various different integration algorithms have been developed for the solution of this equation.
A very efficient one was developed by Ermak and McCammon in 1978 [46, 47]. Here the Brownian equation is solved using a displacement equation of the form:

$$r_i = r_i^0 + \sum_j \frac{\partial D_{ij}^0}{\partial r_i} \triangle t + \sum_j \frac{D_{ij}^0 F_j^0}{kT} \triangle t + R_i(\triangle t) \tag{1.7}$$

with D as the diffusion tensor, F as the force, r as the distance, $\triangle t$ as the size if the time step, T as the temperature and k as the Boltzmann constant.. The variable R is a Gaussian-distribution function and serves as a random displacement with an average value of zero. For further information on integration algorithms for the Brownian equation of motion the reader is referred to [48].

1.2.3. Newtonian molecular dynamics

The most common way to describe the time evolution of a system is by classical concepts, e.g Newton's equation of motion, which can be written as

$$F_i = m_i a_i = m_i \ddot{r}_i \tag{1.8}$$

Here F_i denotes the force acting on the particle i, m_i is the mass, a_i the acceleration and $\ddot{r}_i$ denotes the second derivative of the particle position r with respect to time. The force can be determined by the first derivative of the potential energy function $U(r)$, as a function of the atomic coordinates

$$F_i = -\nabla_i U(r) \ = m_i \ddot{r}_i \tag{1.9}$$

This equation is a second order differential equation. If a Hamiltonian description is used, this equation can be written in a more general way. This is

$$\dot{r_k} = \frac{\partial H(r,p)}{\partial p_k}, \tag{1.10}$$

$$\dot{p_k} = -\frac{\partial H(r,p)}{\partial r_k}. \tag{1.11}$$

If we use the following definition of the Hamiltonian

$$H = H(\mathbf{r},\mathbf{p}) = K(\mathbf{p}) + U(\mathbf{r}) = \sum_i \frac{p_i}{2m_i} + U(\mathbf{r}) \tag{1.12}$$

and plug it into equation 1.10 and 1.11 we recover equation 1.8. In equation 1.12 $K(\mathbf{p})$ denotes the kinetic energy, $U(\mathbf{r})$ the potential energy, p_i the momentum of particle i and m_i the mass of particle i. Newton's equations have several important properties:

- **Conservation of energy** If one assumes that the derivative $\partial H/\partial t = 0$ hols true, the total derivative is also zero. This means Newtons' equations conserve the total energy of the system.

- **Conservation of linear and angular momentum** If no external field is applied, the potential U depends only on the separation of the particles. Under this circumstances the equations also conserve the angular and linear momentum of the system.

- **Time reversibility** If one changes the sign of all the velocities in the system, the system will exactly retrace it's current trajectory. If the equations can be solved exactly, also the numerical trajectory has this property. In practice however, due to the chaotic nature of molecular systems, this reversibility can only be achieved over short time periods.

Since the equation is a second order differential equation it has to be solved by numerical methods. One of the standard methods to solve such problems is the finite-difference approach. Here the coordinates and velocities of the system at the next time step $t+\triangle t$ are calculated from the time and positions of the previous time step which means the equations are solved in a step by step manner. A crucial point is the choice of the step size $\triangle t$ which strongly depends on the types of motion present in the system. As a general rule of thumb, it should be half the timescale of the fastest motion in the system. In order to solve equation (1.8) with a finite-difference approach, let us start with a Taylor expansion:

$$r(t+\triangle t) = r(t) + \dot{r}(t)\triangle t + \frac{1}{2}\ddot{r}(t)\triangle t^2 + \mathcal{O}(\triangle t^3) \tag{1.13}$$

Rewriting the time derivatives one gets

$$r(t + \triangle t) = r(t) + v(t)\triangle t + \frac{1}{2}a(t)\triangle t^2 + \mathcal{O}(\triangle t^3) \tag{1.14}$$

with $v(t)$ as the velocity and $a(t)$ as the acceleration. After discretization of (1.14) one gets

$$r_{n+1} = r_n + v_n\triangle t + \frac{1}{2}\left(\frac{F_n}{m}\right)\triangle t^2 + \mathcal{O}\left(\triangle t^3\right) \tag{1.15}$$

where r_{n+1} denotes the position at time step $t + \triangle t$, r_n the position at time t, $\triangle t$ denotes the step size, F the force acting at time step n and m the mass. The velocity at time $t + \triangle t$ can then be approximated by

$$v_{n+1} = (r_{n+1} - r_n) \,/\, (2\triangle t) \tag{1.16}$$

Using equations (1.15) and (1.16) one gets an integration algorithm with which one can estimate the positions and velocities at time $t+\triangle t$. Since this is a very crude formulation it results in large errors and is just a rough estimate. Using the same arguments as above, one can come up with a far more accurate and sophisticated solution of integrating Newton's equations. Several of these algorithms have been developed up till now and are commonly used in molecular dynamics calculations.
One of the most common integrators is the Verlet algorithm developed by Verlet in 1967 [49]. Verlet used for his scheme two Taylor expansions, one for the forward and one for the backward step

$$r_{n+1} = r_n + v_n\triangle t + \frac{1}{2}\left(\frac{F_n}{m}\right)\triangle t^2 + \mathcal{O}\left(\triangle t^3\right) \tag{1.17}$$

$$r_{n-1} = r_n - v_n\triangle t + \frac{1}{2}\left(\frac{F_n}{m}\right)\triangle t^2 - \mathcal{O}\left(\triangle t^3\right) \tag{1.18}$$

By combining the two equations one gets

$$r_{n+1} = 2r_n - r_{n-1} + \frac{F_n}{m}\triangle t^2 + \mathcal{O}(\triangle t^4) \tag{1.19}$$

$$v_n = \frac{r_{n+1} - r_{n-1}}{2\triangle t} + \mathcal{O}(\triangle t^2) \tag{1.20}$$

The integration for the positions gives quite accurate results with an error estimate of just $\mathcal{O}(\triangle t^4)$. Furthermore, it is independent of the velocity integration. For a whole propagation only one force evaluation is needed, and since the algorithm is defined

in a forward and backward manner it guarantees time reversibility. On the contrary the velocity integration has some rather large errors in the order of $\mathcal{O}(\triangle t^2)$ and can only be calculated if the new positions are already known. Furthermore, the velocities are needed for the kinetic energy calculations which is then also subject to larger errors. Another point is that the algorithm cannot be self started and another Taylor expansion is needed for it's initialization. In order to get rid of the disadvantages the Verlet equations have been modified to a half step scheme which is called the leap-frog algorithm [50, 51], where discretization steps are taken at $n \pm 1/2$

$$r_{n+1} = r_n + v_{n+1/2}\triangle t \tag{1.21}$$

$$v_{n+1/2} = v_{n-1/2} + \frac{F_n}{m}\triangle t \tag{1.22}$$

The current velocities can be calculated according to

$$v_n = \left(v_{n+1/2} - v_{n-1/2}\right)/2 \tag{1.23}$$

The biggest improvement is found in the velocity description since the direct evaluation allows the use of temperature controlling algorithms. Furthermore, the error margin is reduced. On the other hand the velocities are still not handled in a satisfactory manner since they still have to be approximated, and the algorithm is computationally a little more expensive than the Verlet scheme. Another variant, which takes care of the velocity problem, is the velocity Verlet algorithm [52]. A distinct feature is that velocities, positions and accelerations are all stored separately at the same time. In its general form it can be written as

$$r_{n+1} = r_n + v_n\triangle t + \frac{1}{2}\left(\frac{F_n}{m}\right)\triangle t^2 \tag{1.24}$$

$$v_{n+1} = v_n + \frac{1}{2}\left[\frac{F_n}{m} + \frac{F_{n+1}}{m}\right]\triangle t \tag{1.25}$$

The advantages of this algorithm include a great numerical stability, it is easy to implement and it provides direct access to the velocities and therefore, to the kinetic energy. Details on the implementation are given in Section 4.3.2.

1.3. Sampling in different ensembles

A molecular dynamics simulation, as described in the above chapter, can be used to simulate the natural time evolution of a classic system consisting of N particles in

a volume V. This corresponds to a microcanonical ensemble (NVE) where the total energy E, the volume V and the number of particles N are constant. This means that the system has to be completely isolated and unable to exchange energy or particles with the environment. The practical use of the NVE simulations although is limited because it does not correspond to any experimentally realistic situation. Of more practical use are simulations in the canonical (NVT) or isothermal-isobaric (NPT) ensembles, which correspond to the Helmholtz or Gibbs free energy respectively. In the following two sections some of the common simulation techniques for NVT and NPT ensembles are described.

1.3.1. Canonical Ensemble (NVT)

In the canonical ensemble (NVT) the number of particles N, the volume V and the temperature T are kept constant. In contrast to the microcanonical ensemble the energy of the system is not known exactly. The canonical ensemble can describe a closed system in contact with a heat bath. In the context of molecular simulations, the canonical ensemble provides a possibility to perform simulations in a more realistic manner, namely with constant temperature. To measure the temperature in a MD simulation it needs to be expressed in terms of positions and momenta of the particles. A convenient definition of the temperature in a classical many-body system makes use of the equipartition of energy over all degrees of freedom that enter quadratically in the Hamiltonian of the system. In particular for the average kinetic energy per degree of freedom α, we have

$$\left\langle \frac{1}{2} m v_\alpha^2 \right\rangle = \frac{1}{2} k_B T \tag{1.26}$$

with m as the particle mass, v the velocity, k_B as the Boltzmann constant and the temperature T. The brackets $\langle\rangle$ denote the average. In a simulation this equation can be used as an operational definition of the temperature. In practice one would measure the total kinetic energy of the system and divide this by the number of degrees of freedom N_f. As the total kinetic energy of the system fluctuates, so does the instantaneous temperature

$$T(t) = \frac{1}{k_B N_{dof}} \sum_{i=1}^{N_{dof}} m_i \left| v_i^2 \right|, \tag{1.27}$$

where N_{dof} represents the unconstrained degrees of freedom in the system which add up to $3N - n$ (N as number of atoms and n as number of constraints). Thereby k_B is the Boltzmann constant and m and v the mass and velocity of the particle. The simplest way to control the temperature of the system is by scaling the velocities with a

simple scaling factor $\lambda = \sqrt{T_0/T}$ [53] with T_0 as the desired temperature. This method can be used for the heating phase of a MD, but does not reproduce any form of a correct ensemble. A more sophisticated approach to velocity rescaling was developed by Berendsen [54, 55], who coupled a heat bath to the system. The scaling factor he used is written as

$$\lambda = \left[1 + \frac{\triangle t}{2\tau_t}\left(\frac{T_0}{T} - 1\right)\right]^{1/2} \tag{1.28}$$

Here τ_t is the relaxation time which is characteristic for the heat coupling, $\triangle t$ is the time step used in the integration algorithm, T is the instantaneous temperature and T_0 the desired temperature. The drawback of this method is that it does not reproduce a correct canonical ensemble. Another method was proposed by Andersen [56, 57] who used a stochastic force approach. The coupling is realized by adding impulsive forces to randomly selected particles. Between the addition of these forces the system is sampled in the standard microcanonical ensemble. So instead of scaling each velocity at each time step the probability for a particle for a collision is calculated. If the particle is selected for a collision a new velocity is computed from a maxwellian velocity distribution based on the temperature T_0.

One of the most commonly used methods for a NVT simulation is the approach made by Nóse [58]. In his formulation Nóse introduced an additional degree of freedom to the system and rewrote the Langrangian equation of motion for the extended system. This formulation of a heat reservoir allows a dynamic flow of energy from and to the heat bath. In the current implementations the Nóse thermostat is used in the formulation of Hoover [59] which is

$$H_{Nose} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i s^2} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + \frac{g}{\beta} ln\ s \tag{1.29}$$

Thereby s is an additional coordinate added to the system, p are the conjugate momenta, $\beta = 1/k_B T$ with k_B as the Boltzmann constant and T as the temperature, Q is an effective mass and g is related to the degrees of freedom in the system. If this extended system is sampled in a microcanonical ensemble, the sampling of the "real" system corresponds to a canonical ensemble. A critical parameter when this thermostat is used is the effective mass Q. If the value for Q is too high, the energy flow between heat bath and system may become too slow. For a value $Q \longrightarrow \infty$ the microcanonical ensemble is reproduced. If Q is set too small, the energy will oscillate and the equilibration will be slow. Details on the derivation and implementation of the Nóse-Hoover thermostat can be found in section 4.3.3.

1.3.2. Isothermal-Isobaric ensemble (NPT)

The idea of coupling can be further extended to not only influence the temperature but also the pressure of the system. This leads to the isothermal-isobaric ensemble. The isothermal-isobaric ensemble is the most widely used since most experiments are carried out under the conditions of constant temperature and pressure. A condition for pressure coupling is the use of periodic boundary conditions, since the pressure is also dependent on the volume of the simulation box and the volume must be allowed to fluctuate. This fluctuation of the system size is common to all currently applied methods for pressure coupling. One of the first algorithms that coupled the system to the volume of its simulation box was proposed by Andersen [56]. This approach can be seen as a piston acting on a real system. The equations of motion for the extended system can then be written as

$$\dot{r}_i = \frac{P_i}{m_i} + \frac{1}{3}\left(\frac{\dot{V}}{V}\right) r_i \tag{1.30}$$

$$\dot{p}_i = F_i - \frac{1}{3}\left(\frac{\dot{V}}{V}\right) p_i \tag{1.31}$$

$$\ddot{V} = \frac{1}{M_v}\left[P(t) - P_0\right] \tag{1.32}$$

with V as the volume, $P(t)$ the instantaneous pressure, P_0 the desired pressure and F_i, m_i, p_i and r_i as the force, mass, momentum and position of the particle i respectively. Andersen's method could be proven to generate trajectories in the NPH ensemble. In this case the number of particles, the pressure and the enthalpy of the system are conserved. Berendsen [54] proposed an alternative to Andersen's method, which relies on the rescaling of atomic coordinates. The scaling factor for the coordinates and box lengths is

$$\mu = \left[1 - \frac{\triangle t}{\tau_p}\left(P_0 - P\right)\right]^{1/3} \tag{1.33}$$

where $\triangle t$ is the integration timestep, P_0 the pressure of the external pressure bath and τ_p a relaxation time. The instantaneous pressure for the determination of the scaling factor can be derived from the virial theorem [60]

$$P = \frac{2}{3V}\left[E_k + \frac{1}{2}\sum_{i<j} \mathbf{r}_{ij} \cdot \mathbf{F}_{ij}\right] \tag{1.34}$$

Here V is the volume of the simulation box, E_k the kinetic energy, $\mathbf{F}_{ij}$ the force acting between particles i and j and $\mathbf{r}_{ij}$ their respective distance. In standard simulations the simulation box has cubic form, and this form is also retained during the scaling process. The extension to non-cubic box forms, where angles and box lengths can vary independently, has been proposed by Parrinello and Rahman [61] as well as Nóse and Klein [62, 63]. Further constant pressure mechanisms were proposed by Feller [64, 65] and Evans [66]. Details on the implementation of the Berendsen barsotat are given in Section 4.3.3.
Pressure coupling can also be combined with a suitable temperature controlling mechanism which results in the NPT ensemble. One possibility is the combination of the scaling methods, both for temperature and pressure. Another one is the use of the Noóse-Hoover thermostat with the Andersen pressure coupling methods. This combinations yields the best result for NPT calculations. Furthermore, the Langevin piston method can be coupled to a Nóse-Hoover thermostat to obtain a NPT ensemble.

1.4. Free energy calculations

The understanding of chemical processes is mostly glued to the examination of the underlying free energy behavior. Important examples are protein-ligand binding and drug partitioning across cell membranes. Without precise knowledge about the changes in free energy, these processes cannot be understood. Central to the accurate determination of free energy differences between two systems is to explore the configurational space of the reference system, such that relevant low energy states of the target system are adequately sampled. It has been known for a long time that the direct application of conventional computer simulation techniques such as MC or MD are not successful in this respect.

Early development
The solution for this problem came in the 1960s and 1970s when simulation methods were developed which are based on non-Boltzmann sampling. A cornerstone was the energy distribution formalism in which the free energy difference was represented in terms of a one-dimensional integral over the distribution of potential energy differences between the target state and reference state, weighted by an unbiased or biased Boltzmann factor. In 1967 this idea was used by Konrad Singer [67, 68] to calculate the thermodynamic properties of Lennard-Jones fluids. John Valleau and Damon Card [69] devised the so called multistage sampling which relies on the construction of chains of

configurational energies that bridge the reference and target states whenever their low energy regions overlap. This was the birth of the nowadays widely used stratification scheme, where the total free energy difference is split into a sum of free energy differences between intermediate states that overlap considerably better than the initial and final states. Later on Charles Bennet developed an acceptance ratio estimator [70] which corresponds to the minimum statistical variance. The efficiency of this estimator is proportional to the extent to which the two ensembles overlap. Another approach to increase the efficiency of the calculations is to sample the reference state sufficiently broad. If that is the case adequate statistics about the low energy states of the target state are obtainable. A variation of this method was developed in 1977 by Torrie and Valleau [71, 72] who introduced a non-Boltzmann weighting function. Their method became widely known under the name Umbrella Sampling (US). A detailed description of the Umbrella Sampling scheme is given in Section 3.4.1. Due to the low computational power during the time when all these methods were developed, most simulations were based on Monte Carlo simulations. Over the time the calculations were extended from simple Lennard-Jones fluids [68] to atomic clusters [73] and the hydration of ions in small water clusters [74]. Atomic clusters were also the system of interest in one of the first MD free energy studies [75]. All those calculations were based on the early work of Kirkwood [76, 77] the father of the thermodynamic integration (TI) method. In his work he extended the concept of degree of evolution of a chemical system, devised by Theophile De Donder [78] and introduced the order parameter to deduce the free energy difference between two thermodynamic states. His method was used by Mihaly Mezei [79, 80] to calculate the free energy of liquid water. Using a different approach based on multistage and US methods, Patey and Valleau [81] derived a free energy profile for the interaction of an ion pair in a dipolar fluid. At the end of the 70s, free energy calculations were used by several groups to study the free energy of the hydrophobic effect. In 1979, Okazaki [82] used MC simulations to estimate the free energy of hydrophobic hydration and Berne [83] successfully used a multistage strategy to recover results proposed by Patt and Chandler [84]. Further insight was given by calculations of Postma [85] who investigated the solvation of noble gases and was able to estimate the reversible work to form a cavity in water.

Approaching chemically relevant systems

In the 1980s the next step was taken. Lee and Scott [86] devised a method which is nowadays called simple overlap sampling (SOS) and estimated the interfacial free energy of water with MC simulations. A few years later, Warshel was able to calculate the contribution of solvation free energy to electron and proton transfer reactions [87]. A first successful application of a perturbation formalism was realised in 1984 by McCammon [88] on ligand-receptor systems. Based on McCammons success Jorgensen was the first to derive pKa values of simple organic compounds in aqueous solution [89, 90]. These successes marked the turning point for the application of free energy calculations to relevant chemical systems. One of the first complete reaction profiles along an order parameter was obtained by Chandrasekhar [91] by using an Umbrella Sampling approach on a S_N2 reaction of $Cl^- + CH_3Cl$ in water and in vacuum. At the end of the 1980s Tobias and Brooks showed, that the same result can be obtained using a thermodynamic perturbation theory [92]. Significant impact was made in 1987 when Peter Kollman used the free energy perturbation FEP formalism to calculate the free energy change of the transformation between two amino acids [93]. A detailed description of the FEP formalism is given in Section 3.4.2. Furthermore, they were able to derive the binding free energy of thermolysin [94] and substilisin [95] in a protein-inhibitor complex. This indicated that some time in the future it may be possible to model biologically relevant systems.

Reliability of the methods

Although the early calculations showed promising results which were almost in perfect agreement with experimental data, it was realized that some of these early successes were due to luck or fortune rather than adequate and precise methodologies. In some cases it was observed that the free energy deviated from the expected result if the sampling was extended. Careful investigations showed that these algorithms were extremely slow convergent, sometimes to the degree that the system under investigation appeared non ergodic. Based on those observations efforts were made to further improve the common sampling techniques like TI, US, FEP and the stratification scheme. One of the most common problems is the calculation of a free energy profile along a reaction coordinate or order parameter. This type of calculation largely depends on the ability to design bias potentials which enhance sampling. Initial guessing of such potentials is not always easy. Another question is how to get a proper scheme for combining the data of different simulations at different points of the order parameter. A first answer was given by Kumar and others [96–98] with the weighted histogram analysis method (WHAM), for details see Section 3.4.1. Their work was used by Karplus [99] for his

adaptive Umbrella Sampling technique which refined the biasing potential during the course of the simulation. A further problem which was observed when the stratification scheme was used, was the occurrence of singularities at the end points of the transformation. A solution was found by Beutler et al. with the introduction of a softcore potential [100], for details see Section 4.4.2. The use of a softcore potential is nowadays one of the most famous approaches to increase stability and convergence of free energy calculations. Another problem was the dependence on the system size whenever electrostatic interactions were present. The introduction of Ewald like summation [101] strongly diminished this problem. For details on Ewald like summation see Section 4.2. It was not however till Hummer [102] that the size dependence could also be largely eliminated if the simulation cell was charged and not neutral. Unexpected problems arose when holonomic constraints, for details on holonomic constraints see Section 4.3.4, were used during free energy calculations. These constraints are normally used to remove high-frequency vibrations to allow a larger time step during simulations. Karplus and Boresch first proposed a metric tensor correction [103] but the foundations for a correct treatment were laid by Fixman [104] and Scheraga [105] about 20 years later. The complete treatment of those constraints in free energy calculations was then proposed almost another decade later by Otter and Briels [106, 107] who also extended it to the multidimensional case.

Modern algorithms

Based on the solutions for many of the early problems several new algorithms, combining one or more ideas of the early developments emerged. Darve and Pohorille combined the TI and the US approach into a highly efficient adaptive biasing method (ABF) [108, 109]. Compared to the US scheme the ABF is more efficient since it evaluates forces instead of probabilities. The forces are local properties and can be estimated without the need to sample a broad range of the order parameter. The efficiency was shown by Henin and Chipot [110, 111]. ABF proved to provide an almost optimal sampling of the underlying PES even if large barriers are present. Inspired by their work Parrinello and Laio introduced the metadynamic scheme [112] in 2002. It is based on the definition of collective variables to which Markovian dynamics were applied. During the last decade Jarzynski demonstrated the equivalence of the free energy change and an exponential average over the work W along non-reversible paths originating from a canonical ensemble [113, 114]

$$e^{(-\beta \triangle A)} = \left\langle e^{-\beta W} \right\rangle \tag{1.35}$$

Here $\triangle A = A(1) - A(0)$ denotes the free energy difference between the two states. This

can be exploited in practical simulations by moving a constraint on the reaction coordinate relatively fast from an equilibrated system to the target system. This method became known as fast growth [115]. It is related to Bennet's acceptance ratio method. The changes in the energy along these paths are averaged according to equation (1.35). The computational tradeoff is that the faster the constraint is varied, the larger is the statistical spread, and thus, more trajectories have to be calculated. Nevertheless, his work laid the foundation for a whole new class of algorithms to estimate free energies of systems which are either irreversible or driven out of equilibrium. One of the most advanced applications of Jarzynskis identity was done by Schulten and coworkers, who coupled steered MD with Jarzynskis identity to derive free energy profiles for the glycerol conduction in the aquaglyceroporin GlpF [116, 117]. In a more recent work William Jorgensen employed the FEP formalism to improve lead optimization for agents against the HIV-1 virus [118]. Pearlman and Charifson further optimized the efficiency of FEP calculations by suggesting a one-step FEP on a grid surrounding the solute of interest [119].

2. Aim of the work

The aim of this work is the development and implementation of new simulation possibilities for the CAST program package. The CAST program has been in development for a couple of years in the group of Engels and is written in object oriented C++. The purpose of the program is the treatment of large and flexible (macro)molecules. It features several global optimization routines, especially the newly developed TabuSearch [120,121] method as well as standard techniques like Monte Carlo and Monte Carlo with minimization (see Section 1.1). For the underlying energy and gradient calculations a variety of force fields and an interface to the semi empirical program MOPAC exists.

Up until now CAST calculations were performed in serial on a single CPU. To increase the performance and allow the treatment of larger molecules the program is to be modified to support the current multi core architecture of modern computers. In the course of this modification the most time consuming parts of the algorithms, in this case the non-bonded energy and gradient evaluation in the force fields, are to be parallelized using the OpenMP directive. Another bottleneck for calculations using force fields is the build up of the non-bonded list if a cutoff radius is used. To also improve the performance of this step, the build up should be improved by implementation of a linked cell algorithm. Furthermore, the electrostatic treatment of the force fields should be improved by implementation of a smooth particle mesh Ewald summation. The parallelization will be tested on several systems of varying size for a different number of processors and simulation types.

With the increase in computational power, simulations, especially molecular dynamics (MD) simulations and the algorithms based on MD, play an increasing role in the examination of chemical systems. Therefore, CAST will be extended by a molecular dynamics code. The MD part of CAST will feature two different integration schemes for the equations of motion and the possibility to control the temperature and the pressure of the simulation. Control of pressure and temperature implies the underlying control of the simulation boundaries. Therefore, the most common boundary types used in simulations, spherical and periodic boundary conditions, will be implemented. Furthermore, the ability to run constraint simulations will be added by means of an

algorithm based on Lagrange multipliers.
Over the last decade the relevance of free energy calculations has increased dramatically. Especially in the area of protein ligand interactions, these types of calculations play an important role. To accommodate this trend, two free energy methods (umbrella sampling and free energy perturbation) will be implemented in CAST. Both are based on the generation of an ensemble of structures generated for example by molecular dynamics. The methods will be tested against some well known literature examples and an application for the calculation of relative binding free energies will be presented.
One drawback of force fields is the inability to model bond breaking or bond making processes, for example proton transfers in protein-ligand interactions. This limits the application of the optimization and simulation algorithms in CAST. To overcome this drawback, *ab-initio* methods could be used for energy and gradient calculations. To further extend the possibilities of CAST, a MPI interface to the GPU-accelerated DFT code TeraChem should be developed. To investigate the applicability of the CAST/TeraChem interface, conformational search studies on water clusters and a small peptide will be performed.
Next to the direct additions in the CAST program, several smaller programs will be developed to support CAST execution or input file generation. For the easier setup of the chimeric systems used in FEP calculations, two supplementary programs will be designed to aid in structure generation. Furthermore, a conversion program to easily port CHARMM parameters to CAST readable format is presented. The program features the additional possibility to merge two structures and their parameter sets.

3. Theory

3.1. Overview

To successfully study a chemical system using computational methods, a reliable mathematical model for the calculation of the (potential) energy of the system is necessary. The energy is normally described as a function of the structure of the given system. Depending on the question and especially on the size of the system, various different possibilities for these type of calculations exist. For small systems, perhaps also studied in the gas phase, quantum chemical approaches can be used. In this field it can be distinguished between wave function based methods originating from the Hartree-Fock (HF) method [122, 123] and density based methods like density functional theory (DFT) [124]. Today's wave function based methods have all been developed by systematically improving the HF method. If the system size starts to grow, these mathematical formulations rapidly become rather expensive. Here, density functional theory comes into play. Based on the electron density and not on the $3N$ coordinates like the HF and post-HF methods, DFT excels in its performance to calculate systems up to 100 atoms. A major drawback though is the need of a functional, since a proper way for a systematic improvement of functionals doesn't exist. A faster, yet less accurate approach are the semi-empirical methods. The scaling of HF is roughly N^4 with N as the number of basis functions. The main reason for this is the large number of two electron integrals. Semi-empirical methods reduce the number of these integrals by making use of the Zero Differential Overlap [125] (ZDO) assumption. It neglects all products of basis functions which are dependent on the same electron coordinate. By doing so the calculations speed is boosted, but the quality of the wave function diminishes. To make up for this, parameters are introduced. The choice and fitting of the parameters can be done in multiple ways, resulting in different semi-empirical methods like PM3 [14, 126], PM6 [127], AM1 [128] and others. Unfortunately, biomolecules are normally macromolecules including up to several thousands of atoms. If the condensed phase atoms, where the macromolecule is located in, are also taken into consideration, system sizes of more than 20.000 atoms are not uncommon. In this case only empirical

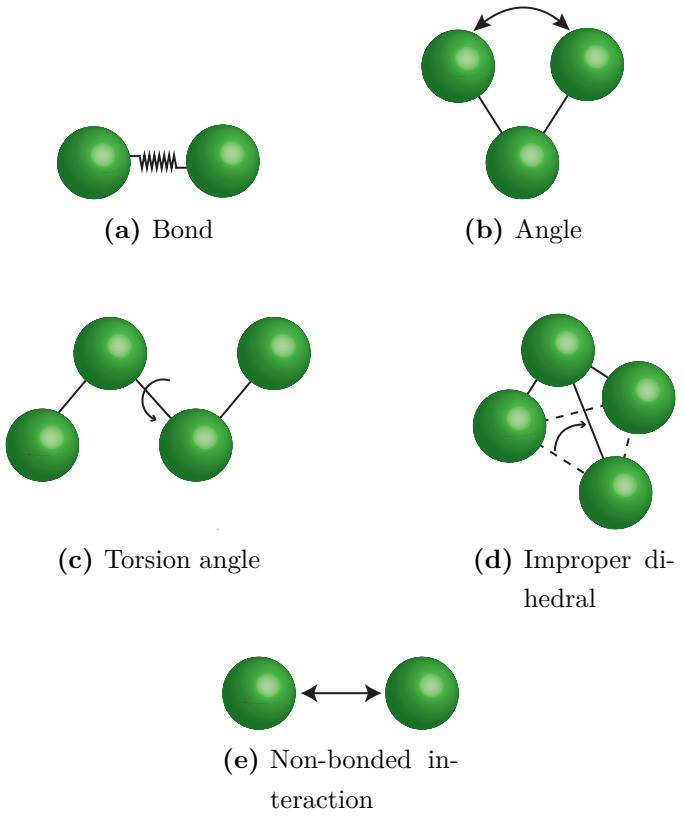
energy functions (force fields) provide the required speed for the calculations. Within force fields the energy E is written as a parametric function, thereby circumventing the above mentioned approximation and solving the Schrödinger equation. A major drawback of this method is the need to fit the functions to either experimental data or high level computations. Due to this, force fields normally are only applicable for a specific group or type of molecules, for example amino acids. Within the force field description the system is decomposed into different subunits, which are assumed to be chemically or structurally equivalent.

3.2. Force fields

The smallest subunit in force fields, for which parameters have to derived, is the atom itself. If only one parameter exists, for example for a carbon atom, regardless of for example the hybridization (sp, sp2, sp3...), it is called a united atom (UA) model. In contrast the all atom (AA) model provides different parameters for carbon atoms, based on hybridization or the corresponding bonding partners or functional groups. Currently several different all atom force fields are commonly used. The general aspects of force fields are illustrated using the OPLS force field. This force field belongs to the class I force fields. In this class the energy is composed only of bonded terms (bonds, angle, torsion angle, improper torsion) and the van-der-Waals and Coulomb non-bonded terms. Other force fields in this class are AMBER and CHARMM. Usually this description is sufficiently accurate. Improvements can be made by adding cross terms (e.g angle-bend) like in the Allinger force fields [22–24] (MM2, MM3...), or enhancing the description of the van-der-Waals and electrostatic term by including multipoles (e.g. AMOEBA). These force fields are commonly referred to as class II and III force fields. The energy of class I force fields can be written as follows:

$$E_{FF} = E_{bond} + E_{angle} + E_{torsion} + E_{improp} + E_{vdW} + E_{coulomb} \tag{3.1}$$

The different contributions to the energy in (3.1) are depicted in Figure 3.1. When comparing AMBER, CHARMM and OPLS, most terms are identical. Small deviations exist in the mixing rules for parameters or calculation of some potentials but essentially they are the same. In the following paragraphs a detailed description of the single potentials is given.



(a) Bond **(b)** Angle **(c)** Torsion angle **(d)** Improper dihedral **(e)** Non-bonded interaction

Figure 3.1.: Picture of the different contributions to the total force field energy. Pictures (a-d) compose the different bonded interaction whereas (e) illustrates the non-bonded interactions

Bonding energy [122]

The bonding energy results form the stretching motion of a bond between two adjacent atoms and can be derived starting from a Taylor expansion around the equilibrium bond distance

$$E_{bond}\left(r^{AB}-r_0^{AB}\right) \approx E(0)+\frac{dE}{dr}\left(r^{AB}-r_0^{AB}\right)+\frac{1}{2}\frac{d^2E}{dr^2}\left(r^{AB}-r_0^{AB}\right)^2 \tag{3.2}$$

Here the expansion is cut after the second order term. The $E(0)$ term is the zero point energy and set to zero. The first order term is zero as well due to the expansion around the equilibrium. Therefore, the bonding energy can be written as

$$E_{bond}\left(r^{AB}-r_0^{AB}\right) = k^{AB}\left(r^{AB}-r_0^{AB}\right)^2 = k^{AB}(\triangle r^{AB})^2 \tag{3.3}$$

Thereby r^{AB} denotes the distance between atoms A and B, k is the force constant and r_0^{AB} the equilibrium distance. This harmonic form is the standard form for class I force fields. One major problem of these potentials lies in the limiting behavior. When the bond is stretched to infinity the energy should not increase beyond the bond dissociation energy. Since the harmonic potential is limitless the energy will reach to $+\infty$. Improvements can be made by changing the harmonic to a Morse potential,

$$E_{Morse}(\triangle r) = D\left[1 - e^{\alpha \triangle r}\right]^2 \tag{3.4}$$

with D as the dissociation energy and $\alpha = \sqrt{k/2D}$, k being the force constant. Problems may arise with this potential if the starting geometry contains elongated bonds. In this case convergence to the equilibrium distance may be slow. Therefore, the common force fields normally rely on the harmonic form of the bonding potential.

Angle energy [122]

The angle energy is the energy resulting from bending an angle between 3 atoms (A-B-C). In this case a bond exists between A-B and B-C. Similar to the bonding energy, the angle potential is expanded in a Taylor series around the equilibrium angle. Truncating this potential again after the second order and using the same arguments as in the bonding potential the energy can be written as

$$E_{Angle}\left(\Theta^{ABC} - \Theta_0^{ABC}\right) = k^{ABC}\left(\Theta^{ABC} - \Theta_0^{ABC}\right)^2 \tag{3.5}$$

Here Θ^{ABC} is the current angle, Θ_0^{ABC} the equilibrium angle and k^{ABC} the force constant for the potential. The accuracy of this harmonic form is sufficient for most calculations. If higher accuracy is needed, third order terms can be included.

Torsion energy [122]

The torsion potential arises due to the rotation around a single bond. Four atoms A, B, C, and D are involved with bonds between A-B, B-C and C-D, see Figure 3.2. The rotation occurs around the B-C bond. The corresponding torsion angle can exert values either from 0° to 360° or −180° to +180° depending on the used nomenclature. There are some major differences between the torsion potential and the before mentioned bond and angle potentials. When looking at Figure 3.2 it becomes clear that the torsion

potential has to be symmetric with respect to the rotation around the B-C bond. This means after a 360° rotation, the energy has to have the same value as at the starting point.

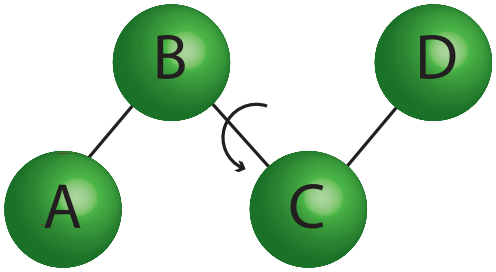


Figure 3.2.: The four atoms involved in a torsion angle. The angle is defined as the angle between planes A-B-C and B-C-D.

Furthermore, rotations around single bonds are common. In return, this means the energy needed for such a rotation is usually pretty low and large deviations from the equilibrium of the torsion angle are possible. To make up for this, the energy is not expanded in a Taylor series but a Fourier series. This ensures the possibility of large deviations, as well as the periodicity of the potential. The energy is approximated by Fourier Series as follows

$$E_{torsion}(\omega) = \sum_{n=1} V_n cos(n\omega) \tag{3.6}$$

Thereby ω is the value of the angle and the constant V_n determines the barrier size for the rotation around B-C. The factor $n = 1, 2, 3, ...$ determines the periodicity. $n = 1$ describes a periodicity of 360°, $n = 2$ a periodicity of 180°. Terms with higher periodicity can be taken into account accordingly. Depending on whether a single or double bond is described the torsion potential has to have several minima. This can be achieved by adding more than one term to the potential:

$$\begin{aligned} E_{torsion}(\omega) &= \frac{1}{2} V_1^{ABCD} \left[1 + cos\left(\omega^{ABCD}\right)\right] \\ &+ \frac{1}{2} V_2^{ABCD} \left[1 - cos\left(2\omega^{ABCD}\right)\right] \\ &+ \frac{1}{2} V_3^{ABCD} \left[1 + cos\left(3\omega^{ABCD}\right)\right] \end{aligned} \tag{3.7}$$

Setting the minima for the different terms is achieved by the + and - signs. The onefold rotational term now has a minimum at 180°, the twofold rotational term at 0° and 180°

and the threefold rotational term at 60°, 180° and 300°. The factor 1/2 ensures that the potential height is recovered if only one term is present.

Improper torsion energy [122]

The improper torsion is defined as the out-of-plane angle of a trigonal planar system where the central atom is sp^2-hybridized, see Figure 3.3. Normally this arrangement is planar and a significant amount of energy is needed for pyramidalization. It can be shown however, that only a small distortion of the in-plane ADB, ADC and BDC angles is needed to break the planarity.

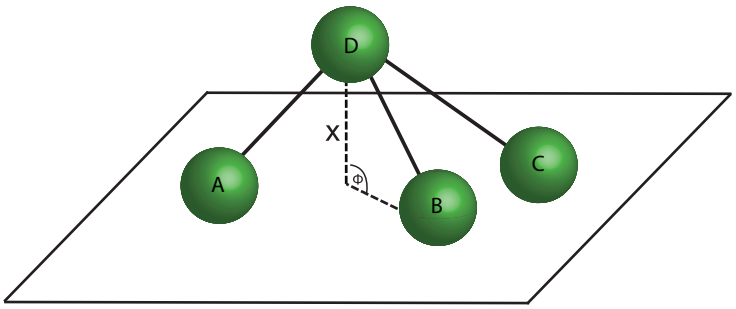


Figure 3.3.: Illustration of the out-of-plane angle Φ.

To make up for this, very large force constants for the in-plane angles had to be used which in return makes the "normal" angles unrealistically stiff. Therefore, these out-of-plane angles are treated separately. The improper potential can be written in a harmonic form:

$$E_{improp}(\Phi^D) = k^D(\Phi^D)^2 \qquad or \qquad E_{improp}(x^D) = k^D x^2 \tag{3.8}$$

Here k denotes the force constant and Φ or respectively x the deviation from the plane.

Non bonded energy terms [122, 129]

van-der-Waals The non-bonded potential can be divided into the van-der-Waals and electrostatic part. Both have in common that they describe interactions between atoms which are not directly bonded. The van-der-Waals term is the part of the interaction which does not arise due to the electric charges of the involved atoms. It is the main interaction energy between alkanes or noble gases. It arises if the two electron clouds of the respective atoms or molecules overlap. At infinite distance the potential is equal

to zero. If the particles get closer to each other the potential becomes slightly negative, until on further approach it becomes extremely repulsive due to the negative charge on both electron clouds. The slight attraction can be seen as dipole-dipole interactions. Since the electrons are "moving" constantly the electron distribution is not uniform. Because of this, due to the uneven distribution a dipole on one particle may build up. This dipole can induce another dipole on the neighboring particle, resulting in an attractive dipole-dipole interaction. This attractive potential changes with the inverse 6th power of the distance between the fragments. The force, resulting from this potential is called "London" or "dispersion" force [130]. One of the most commonly used potentials which describes this behavior rather accurately is the following Lennard-Jones potential [131], see also Figure 3.4.

$$E_{vdW}(r) = \varepsilon \left[\left(\frac{r_0}{r} \right)^{12} - \left(\frac{r_0}{r} \right)^{6} \right] \tag{3.9}$$

where ε denotes the minimum depth, r_0 the minimum distance and r the current distance between particles. The selection of the exponent for the repulsive part is solely up to the user. Normally it is chosen as to the power of 12. This makes it easier for programs to process this part because the attractive part has only to be multiplied with itself. To justify the choosing of the repulsive part as an exponential function instead of $\frac{1}{r^{12}}$ one can refer to electronic structure theory. It is known that the electron density decays exponentially when moving away from the nucleus. Furthermore, the exact hydrogen wave function is exponential. A general form of the van-der-Waals potential would then be

$$E_{vdW}(r) = Ae^{-Br} - \frac{C}{r^6} \tag{3.10}$$

with A, B and C as constants. In a more detailed form it can be written as

$$E_{vdW}(r) = \varepsilon \left[\frac{6}{\alpha - 6} e^{\alpha(\frac{1-r}{r_0})} - \frac{\alpha}{\alpha - 6} \left(\frac{r_0}{r} \right)^{6} \right] \tag{3.11}$$

where α is a free parameter, ε and r_0 are defined as in Equation (3.9). Depending on the value of α, the Lennard-Jones behavior at long distances can be recovered ($\alpha = 12$). Since both ε and r_0 are dependent on two atoms, especially the atom types, there are several ways a force field can handle these parameters. Most force fields use combination rules of the individual atom type parameters to generate the final diatomic parameters.

The OPLS force field for example uses the following rules

$$\begin{aligned} r_0^{AB} &= \sqrt{r_0^A \; r_0^B} \\ \varepsilon_{AB} &= \sqrt{\varepsilon^A \; \varepsilon^B} \end{aligned} \tag{3.12}$$

The total radius r_0^{AB} is the geometrical mean of the van-der-Waals radii of both atoms r_0^A and r_0^B. The ε value is chosen as the geometrical mean of the individual ε_A, ε_B.

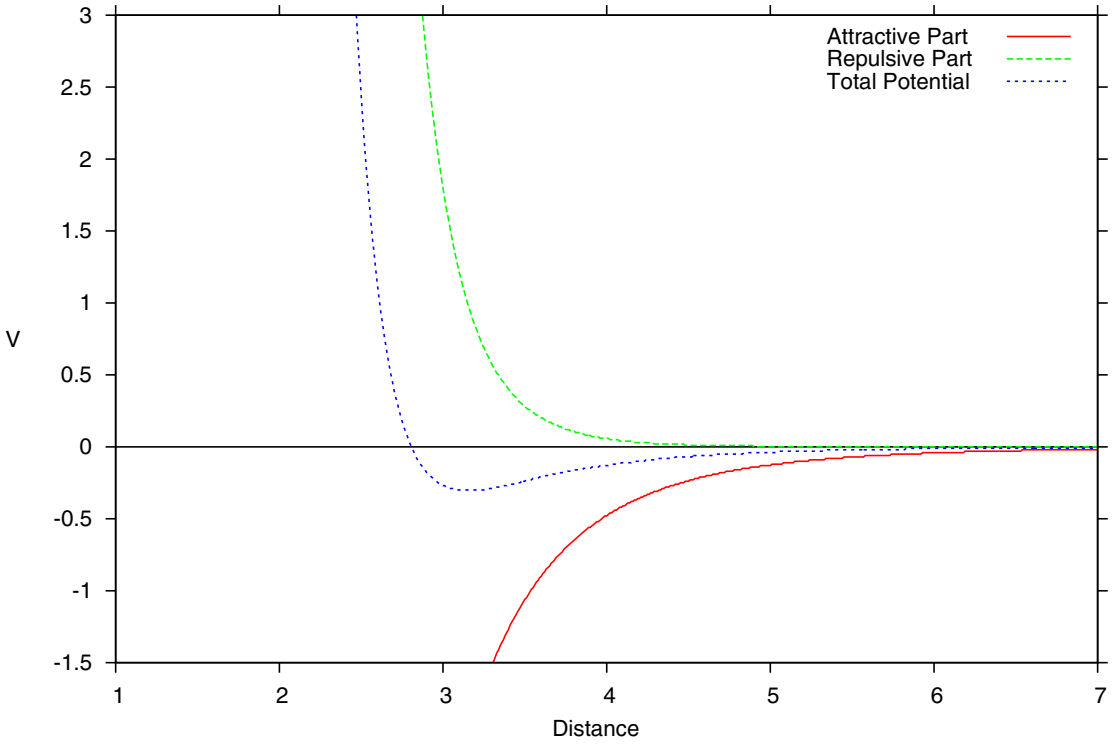


Figure 3.4.: Plot for a Lennard-Jones potential in arbitrary units. The repulsive and attractive part can be seen in green and red. The total potential is shown in blue.

Another approach was developed by Halgren in 1992 [132]. He proposed a buffered 14-7 potential for the van-der-Waals interaction between two paticles i and j

$$V = \varepsilon_{ij} \cdot \left[\frac{1.07}{p_{ij} + 0.07}\right]^7 \cdot \left[\frac{1.12}{p_{ij}^7 + 0.12} - 2\right] \tag{3.13}$$

Here ε_{ij} is the well depth and p_{ij} is defined as R_{ij}/R_{ij}^0 with R_{ij} as the current distance and R_{ij}^0 as the minimum distance between particles i and j. This form is adopted in the AMOEBA force field. It is supposed to provide a better fit to gas phase calculations and the repulsion part is a little bit softer than the Lennard-Jones equivalent.

Electrostatics The second non-bonded interaction type arises due to the charged parts on the atoms created by the internal distribution of electrons. The most simple way to model these charged parts is by assigning partial charges to each atom. This interaction between point charges is then described by the following Coulomb potential

$$E_{el}(r^{AB}) = \frac{Q^A Q^B}{\varepsilon r^{AB}} \tag{3.14}$$

where Q_A and Q_B resemble the atomic charges, ε is a dielectric constant and r the distance between charges. Most class I force fields use this description for the electrostatic energy. The charges are most commonly fitted to the electrostatic potential obtained by calculations using electronic structure methods.
The electrostatic energy can also be extended to include not only monopoles but also dipoles and higher multipoles. The AMOEBA [129] force field uses terms up to quadrupoles.

$$M_i = [q_i, \mu_{ix}, \mu_{iy}, \mu_{iz}, Q_{ixx}, Q_{ixy}, Q_{ixz}, ...Q_{izz}]^T \tag{3.15}$$

with q as the charge at site i, μ the dipole, Q the quadrupole and t denotes the transpose. Energy between two sites can be calculated using the cartesian polytensor formalism [133–135].

$$U_{elec}^{perm}(r_{ij}) = M_i^T \; T_{ij} \; M_j \tag{3.16}$$

where

$$T_{ij} = \begin{pmatrix} 1 & \frac{\partial}{\partial x_j} & \frac{\partial}{\partial y_j} & \frac{\partial}{\partial z_j} \\ \frac{\partial}{\partial x_i} & \frac{\partial}{\partial x_i x_j} & \frac{\partial}{\partial x_i y_j} & \frac{\partial}{\partial x_i z_j} \\ \frac{\partial}{\partial y_i} & \frac{\partial}{\partial y_i x_j} & \frac{\partial}{\partial y_i y_j} & \frac{\partial}{\partial y_i z_j} \\ \frac{\partial}{\partial z_i} & \frac{\partial}{\partial z_i x_j} & \frac{\partial}{\partial z_i y_j} & \frac{\partial}{\partial z_i z_j} \end{pmatrix}$$

Cross terms [122]

Beginning with class II forced fields not only the standard potentials as mentioned above are used, but terms which couple two or more of them. These terms cover combinations of movements in the system which the other terms cannot. For example, if the angle in the H_2O molecule changes, the bond length to the hydrogen atoms also changes. The cross terms which are able to model such combinatorial movements are usually written as products of Taylor expansions. One of the most important terms is the stretch-bend cross term

$$E_{str-bend} = k^{ABC} \left(\Phi^{ABC} - \Phi_0^{ABC}\right) \left[\left(r^{AB} - r_0^{AB}\right) + \left(r^{BC} - r_0^{BC}\right)\right] \tag{3.17}$$

where k^{ABC} is the force constant for the potential, Φ^{ABC} is the angle between atoms A-B-C and r^{AB} and r^{BC} are the distances between atoms $A-B$ and $B-C$ respectively. Other examples include the combination of stretch and torsional, Equation (3.18), or bend and torsional, Equation (3.19), movement

$$E_{str-tors} = k^{ABCD}\left(r^{AB} - r_0^{AB}\right) cos\left(n\omega^{ABCD}\right) \quad (3.18)$$

$$E_{bend-tors-bend} = k^{ABC}\left(\Phi^{ABC} - \Phi_0^{ABC}\right)\left(\Phi^{BCD} - \Phi_0^{BCD}\right) cos\left(n\omega^{ABCD}\right) \quad (3.19)$$

The force constants k^{ABCD} and k^{ABC} involved in these expressions are normally only fitted to few of the atoms involved, for example the central atom in an 3-atom (ABC) situation. Another possibility is to use an universal constant.

3.3. Free energy

3.3.1. Mathematical formulation [136–138]

Density of states One of the key functions in statistical mechanics, and the link between the microscopic and macroscopic properties of a system, is the density of states. As example we consider the density of states Ω of a system of N particles, also called the microcanonical partition function. Ω is here defined as the energy density of microstates. The density of states for a discrete system counts the number of microstate configurations which are consistent to each macrostate. For example $\Omega(\varepsilon)$ returns the number of microstates with energy ε. If the system possesses no discrete degrees of freedom, this counting is no longer possible. The possible number of configurations is equal to infinity. For the considered system we use the complete $3N$-dimensional space, which is defined by the cartesian coordinates of the particles and make the following assumption: the "number" of configurations equal to a specific energy, is proportional to the $(3N-1)$ dimensional hypersurface of the system

$$\Omega_{conf} \propto \frac{1}{N!}\int_{V^N} \delta\left[U(\mathbf{q}) - \varepsilon\right] d\mathbf{q} \quad (3.20)$$

The subscript $conf$ is used since the integral solely depends on the energy and configuration of the system. U refers to the potential energy, q are the $3N$ coordinates, δ is the Dirac delta function and N and V define the dimensionality and the boundaries of the hypersurface. The integral is calculated over the whole volume V^N and the delta

function only returns those configurations which correspond to the energy ε. If two configurations only differ by permutation of one or more particles the configurations are seen as identical and sorted out by the factor $N!$. Taking the Hamiltonian of the system, the complete density of states can be written as:

$$\Omega_{tot}(N,V,\varepsilon) = \frac{1}{h^{3N}N!}\int\int_{V^N}\delta\left[\mathcal{H}(\mathbf{q},\mathbf{p_x})-\varepsilon\right]d\mathbf{q}d\mathbf{p_x} \tag{3.21}$$

The factor $1/h^{3N}$ has been introduced for proportionality reasons, and $\mathbf{p}_x$ corresponds to the $3N$ momenta. The interesting information gathered from simulations of molecular systems normally only include configurational properties. Since the kinetic component is an analytical expression, it can be integrated out if simulations are run for example in the canonical NVT ensemble. For a microcanonical ensemble like that given in Equation (3.21) the situation is different. As kinetic and potential energy in the Hamiltonian are additive the δ-function in Equation 3.21 can be written as a convolution integral with two δ functions

$$\begin{aligned}\Omega_{tot}(N,V,\varepsilon) &= \frac{1}{h^{3N}N!}\int\int_{V^N}\delta\left[U(\mathbf{q})+K(\mathbf{p}_x)-\varepsilon\right]d\mathbf{q}d\mathbf{p_x} \\ &= \frac{1}{h^{3N}N!}\int\left[\int\delta\left[K(\mathbf{p}_x)-\varepsilon^{'}\right]d\mathbf{p}_x\right] \\ &\times \left[\int_{V^N}\delta\left[U(\mathbf{q})-\varepsilon+\varepsilon^{'}\right]d\mathbf{q}\right]d\varepsilon^{'} \\ &= \int\Omega_{ig}(N,V,\varepsilon^{'})\Omega_{ex}(N,V,\varepsilon-\varepsilon^{'})d\varepsilon^{'}\end{aligned} \tag{3.22}$$

where Ω_{ig} and Ω_{ex} are called the ideal gas (ig) and excess (ex) density of states and are defined as follows

$$\begin{aligned}\Omega_{ig}(N,V,\varepsilon) &= \frac{V^N}{h^{3N}N!}\int\triangle\left[K(\mathbf{p}_x)-\varepsilon\right]d\mathbf{p}_x \\ &= \left[\frac{(2\pi m\varepsilon)^{3/2}V}{h^3}\right]^N\frac{\varepsilon^{-1}}{N!\Gamma(\frac{3}{2}N)}\end{aligned} \tag{3.23}$$

$$\Omega_{ex}(N,V,\varepsilon) = \frac{1}{V^N}\int_{V^N}\triangle\left[U(\mathbf{q})-\varepsilon\right]d\mathbf{q} \tag{3.24}$$

where Γ is the Euler Gamma function, m the particle mass and K a quadratic funtion of the momentum $K=\sum\mathbf{p}^2/2m$. The main point of this rewriting is that it shows that the Ω_{ig} expression (the kinetic contribution) of the microcanonical partition function is an analytical expression. During simulations the Ω_{ex} part is the one we are interested in. Comparing the excess density of states with the configurational density of states

shows only a small difference

$$\Omega_{ex}(N,V,\varepsilon) = \frac{N!}{V^N}\Omega_{conf}(N,V,\varepsilon) \tag{3.25}$$

namely the factor $N!/V^N$. Simulation algorithms can now be formulated in a way that either the excess or configurational density of states is calculated. The only difference then is the multiplicative factor which is either incorporated in Ω or used in the reweighting of the results. The physically significant description although, is Ω_{conf} since it is the one proportional to the number of microstates with the corresponding N, V and ε.

Partition function Another key function of statistical mechanics and thermodynamics is the partition function. It describes the statistical properties of a system in thermodynamic equilibrium. Often properties like the total energy, free energy and others are expressed in terms of this function or its derivatives. The partition function depends on the type of statistical ensemble. It is directly related to the density of states by Laplace transformation. The Laplace transformation of a function $F(t)$ is defined as follows

$$\begin{aligned} f(s) = \mathcal{L}F(t) &= \lim_{a\to\infty}\int_0^a e^{-st}F(t)dt \\ &= \int_0^\infty e^{-st}F(t)dt \end{aligned} \tag{3.26}$$

Taking the Laplace transform $\mathcal{L}$ of the density of states (Ω_{tot}) leads to the canonical partition function

$$\begin{aligned} Q(N,V,T) &= \int d\varepsilon \int\int_{V^N} \delta\left[\mathcal{H}(\mathbf{q},\mathbf{p_x}) - \varepsilon\right] e^{-\beta\varepsilon} d\mathbf{q}d\mathbf{p_x} \\ &= \int e^{-\beta\varepsilon}\Omega_{tot}(N,V,\varepsilon)d\varepsilon \end{aligned} \tag{3.27}$$

Replacing the one dimensional integral over energy, with a multidimensional integral over coordinates leads to the most familiar description of the canonical partition function

$$Q(N,V,T) = \int e^{[-\beta\mathcal{H}(\mathbf{q},\mathbf{p}_x)]}d\mathbf{q}d\mathbf{p}_x \tag{3.28}$$

Free energy The free energy is the energy that is needed to generate a system in thermal equilibrium with its surrounding at a defined temperature. Depending on the thermodynamic ensemble the Helmholtz or the Gibbs free energy can be obtained. They

are defined as follows

$$A = U - TS \tag{3.29}$$

$$G = U - TS + pV = H - TS \tag{3.30}$$

with U as the internal energy, H the enthalpy, p the pressure, V the volume, T the temperature and S the entropy. The free energy can be expressed in terms of the canonical partition function, by expressing U and S in terms of the partition function. It can be readily shown that the average energy $\langle E \rangle$ of a system inside an ensemble is the derivative of the logarithm of the partition function Q with respect to β

$$\begin{aligned}
\langle E \rangle &= \sum_j p_j(N,V,\beta) E_j(N,V) \\
&= \sum_j \frac{E_j(N,V) e^{-\beta E_j(N,V)}}{Q(N,V,\beta)} \\
p_j &= \frac{e^{-\beta E_j}}{Q(N,V,\beta)} \\
\left(\frac{\partial ln Q(N,V,\beta)}{\partial \beta}\right)_{N,V} &= \frac{1}{Q(N,V,\beta)} \left(\frac{\partial \sum e^{-\beta E_j(N,V)}}{\partial \beta}\right)_{N,V} \\
&= \frac{1}{Q(N,V,\beta)} \sum \left[-E_j(N,V)\right] e^{-\beta E_j(N,V)} \\
&= \sum_j \frac{E_j(N,V) e^{-\beta E_j(N,V)}}{Q(N,V,\beta)} \\
\Rightarrow \langle E \rangle &= \left(\frac{\partial ln Q(N,V,\beta)}{\partial \beta}\right)_{N,V} = k_B T^2 \left(\frac{\partial ln Q}{\partial T}\right)_{N,V}
\end{aligned} \tag{3.31}$$

The entropy can be expressed in terms of the partition function by using Boltzmann's formula

$$S = k_B \cdot ln W \tag{3.32}$$

with k_B as the Boltzmann constant and W as the weight of a distribution, defined as $W(n_1, n_2, ...n_j) = \frac{N!}{n_1! n_2! ...} = \frac{A!}{\prod_j a_j!}$. Using Sterling's formula [139, 140] and inserting the definition for the probability distribution p_j from Equation (3.31) yields

$$S_{System} = -k_b \sum_j p_j \; ln p_j \tag{3.33}$$

Inserting Q in Equation (3.33) and using the definition of $\langle E \rangle$ from Equation (3.31) it follows

$$\begin{aligned} S &= -k_B \sum_j \frac{e^{-\beta E_j}}{Q}(-\beta E_j - lnQ) \\ &= \frac{U}{T} + k_B\, lnQ \\ &= k_B T \left(\frac{\partial lnQ}{\partial T}\right)_{N,V} + k_B\, lnQ \end{aligned} \tag{3.34}$$

Inserting Equations (3.34) and (3.31) into Equation (3.29) yields the final expression for the free energy in terms of the partition function

$$F = -\frac{1}{\beta}\, lnQ(N,V,T) \tag{3.35}$$

This equation shows, that in order to calculate F one has to estimate the value of Q. The calculation of Q is often not possible or at least extremely difficult. Nevertheless, chemists are normally only interested in the difference ($\triangle F$) of the free energy between two states (for example a and b) of the same system or two distinct systems. These systems can be described by their individual partition functions. Mathematically this can be expressed as

$$\triangle F = -\frac{1}{\beta} ln \frac{Q_b}{Q_a} \tag{3.36}$$

Formula (3.36) implies that the underlying problem of computing free energy differences is the determination of the ratio of the partition functions of the two states. There exist several ways for computer simulations to access this problem. These methods are described in Section 3.4

3.3.2. Ergodicity principle

Ergodicity is a central feature a system has to have if it is to be examined by computer simulations. The time evolution of a N particle system can be seen as a trajectory in phase space with the initial coordinates and momenta $(\mathbf{p_0}, \mathbf{q_0})$. The time average of any observed characteristic f over time T of a system can then be written as

$$\overline{f(\mathbf{q_0}, \mathbf{p_0})} = \frac{1}{T} \int_0^T f\left[\mathbf{q(t)}, \mathbf{p(t)}\right] dt \tag{3.37}$$

and the ensemble average as

$$\langle f \rangle = \int f P(\mathbf{x}, \mathbf{p}_x) d\mathbf{x}\ d\mathbf{p}_x \tag{3.38}$$

where $P(\mathbf{x}, \mathbf{p}_x)$ denotes the probability for systems being in the state $(\mathbf{x}, \mathbf{p}_x)$. For an ergodic system the time average becomes the ensemble average in the limit of infinite simulation time

$$\lim_{T\to\infty} \overline{f(\mathbf{q_0}, \mathbf{p_0})} = \langle f \rangle \tag{3.39}$$

Equation (3.39) is also known as the *ergodic hypothesis*. Two important consequences arise from this Equation if Monte Carlo or molecular dynamics simulations are used to generate system trajectories. Based on Equation (3.39) it becomes clear that averaged properties of the systems can be extracted from these kind of computations. Furthermore, if the simulation time is sufficiently long, the initial conditions $(\mathbf{p_0}, \mathbf{q_0})$ don't influence the results of the calculated averages. Proving ergodicity for a many-body system is usually extremely difficult. Today it is assumed that almost any many-body system is ergodic, although a few non-ergodic systems are known. These include systems where the constants of motion are equal to the number of degrees of freedom. This was proven in the Kolmogorov-Arnold-Moser (KAM) theorem [141].

3.4. Calculating free energy differences [136]

Calculation of free energy differences between two states is normally based on the use of a reaction coordinate q. The reaction coordinate can be any kind of parameter (distance, angle, change in the energy expression, etc.). Referring back to the ergodic principle, a sufficiently long simulation time has to be employed along the coordinate. Nevertheless, the simulation time still is finite for real calculations. Therefore, an efficient sampling algorithm is needed, which also covers high energy regions of the phase space. The currently available methods for such calculations can roughly be divided in three sections. The first kind of methods sample the system in equilibrium. The second class comprises non-equilibrium methods. The third kind of methods add another degree of freedom to the system. Common to many of these methods is the division of the reaction path into segments (windows). For each interval a separate simulation is run. When all simulations have been run, the respective data are combined to retrieve the complete free energy profile. A depiction of the stratification scheme can be seen in Figure 3.5. In Chapter 1 several methods for the calculation of free energy differences were introduced. In the following sections a detailed description of the methods which

were implemented in the CAST program package are given. In the next paragraphs the umbrella sampling as well as the weighted histogram analysis method are described. The section is then closed with the derivation of the free energy perturbation formalism.

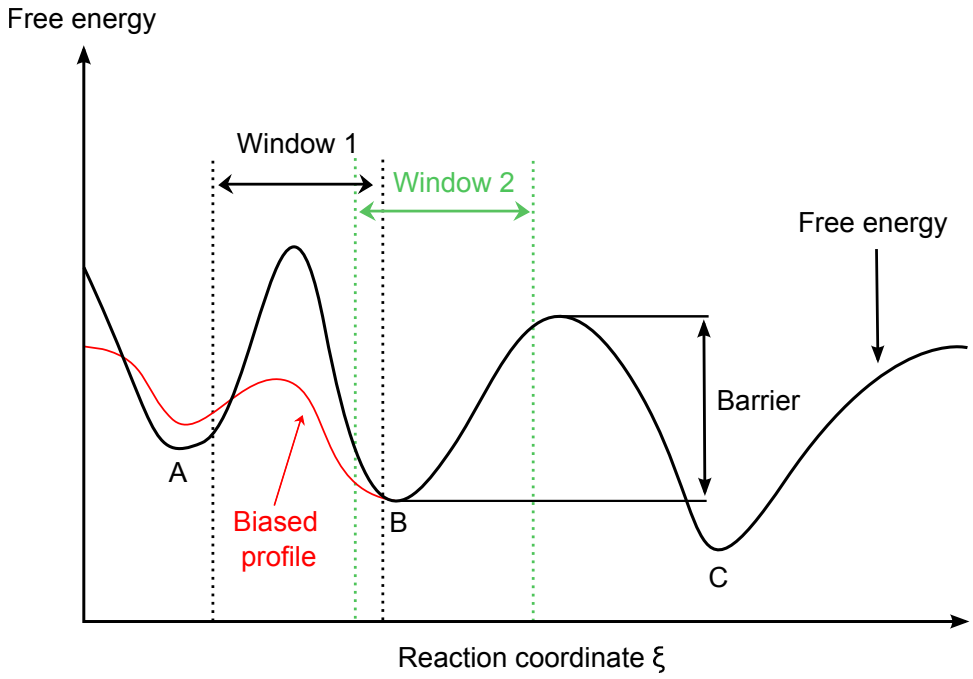


Figure 3.5.: Illustration of the stratification process. The free energy profile is split into several windows. For each window a separate simulation is run. Results of the single simulations are then combined to retrieve the complete free energy profile. The red curve indicates, how a bias potential can "smoothen" the profile and simplify sampling of transition regions otherwise unavailable. Figure reproduced and modified from ref [136] with permission from Springer.

3.4.1. Umbrella sampling [142]

For a reaction coordinate q the probability distribution of a system along q can be calculated by

$$Q(q) = \frac{\int \delta\left[q(r) - q\right] e^{-\beta E} d^N r}{\int e^{-\beta E} d^N r} \tag{3.40}$$

The probability distribution Q can be interpreted as the probability of finding the system in a small interval dq around q. Calculation of the free energy along the reaction coordinate can now be performed via

$$A(q) = -1/\beta \; lnQ(q) \tag{3.41}$$

Unfortunately these direct phase space integrals like (3.40) are extremely difficult or impossible to calculate in computer simulations. If however the system is ergodic, and if during the simulation every point in phase space is visited the probability distribution is equal to

$$P(q) = \lim_{t\to\infty} \frac{1}{t} \int_0^t p\left[q(t')\right] dt' \tag{3.42}$$

In that case the ensemble average $Q(q)$ is equal to the time average $P(q)$ for infinite sampling time. In Equation (3.42) t is the time and p is the number of occurrences of q during the simulation in a given interval. The free energy A can therefore be directly obtained from molecular dynamics simulations by simply monitoring $P(q)$. The problem which can be tackled with such an approach are possible barriers along the reaction coordinate. A normal MD will sufficiently sample the minimum regions of the surface, but the transitions or the barrier regions would be poorly sampled. To be able to use Equation (3.39) the whole phase space has to be sampled, including the barrier regions. One possible way to sample regions which are higher in energy is to alter the energy expression by applying a bias potential. This method is known as Umbrella Sampling (see Chapter 1). For each segment (window) of the reaction coordinate q a separate simulation is run and $P(q)$ is collected. When the simulations are done the biased distributions have to be unbiased and combined to extract the potential of mean force (PMF).
The bias potential ω_i of window i is an additional term which only depends on the reaction coordinate via

$$E^b(r) = E^u(r) + \omega_i(q) \tag{3.43}$$

Here $E(r)$ is the energy of the system and b denotes biased quantities whereas u denotes the unbiased quantities. To obtain the unbiased free energy the unbiased distribution P^u is needed

$$P_i^u(q) = \frac{\int \delta\left[q'(r) - q\right] e^{[-\beta E(r)]} d^N r}{\int e^{[-\beta E(r)]} d^N r} \tag{3.44}$$

If we perform an MD simulation with a biased potential we will obtain the biased distribution

$$P_i^b(q) = \frac{\int \delta\left[q'(r) - q\right] e^{\left[-\beta E(r) + \omega_i(q'(r))\right]} d^N r}{\int e^{\left[-\beta E(r) + \omega_i(q'(r))\right]} d^N r} \tag{3.45}$$

Since the bias depends only on q we can extract it from the integral in the enumerator

$$P_i^b(q) = e^{[-\beta\omega_i(q)]} \frac{\int \delta\left[q'(r) - q\right] e^{[-\beta E(r)]} d^N r}{\int e^{\left[-\beta E(r) + \omega_i(q'(r))\right]} d^N r} \tag{3.46}$$

Writing the unbiased distribution in terms of the biased distribution yields

$$\begin{aligned} P_i^u(q) &= P_i^b(q)e^{[\beta\omega_i(q)]} \cdot \frac{\int e^{[-\beta E(r)+\omega_i q(r))]}d^N r}{\int e^{[-\beta E(r)]}d^N r} \\ &= P_i^b(q)e^{[\beta\omega_i(q)]} \cdot \frac{\int e^{[-\beta E(r)]}e^{[-\beta\omega_i(q(r))]}d^N r}{\int e^{[-\beta E(r)]}d^N r} \\ &= P_i^b(q)e^{[\beta\omega_i(q)]}\langle e^{[-\beta\omega_i(q)]}\rangle \end{aligned} \tag{3.47}$$

With the result from Equation (3.47) the free energy can be calculated. $P_i^b(q)$ can simply be obtained from the MD simulation, ω_i is an analytical expression and the final term $F_i = \frac{-1}{\beta} ln(e^{-\beta\omega_i(q)})$ is independent of q. The expression for the free energy then reads

$$A_i(q) = \frac{-1}{\beta} ln(P_i^b(q)) - \omega_i(q) + F_i \tag{3.48}$$

This derivation is exact. The only approximation made is that our sampling is sufficient. If the whole reaction coordinate is covered in a single window with one simulation Equation (3.48) can be used to calculate the PMF. If more than one window is used one has to find a method that combines the windows.

Weighted histogram analysis method (WHAM) To obtain the PMF from umbrella sampling data, histogram analysis can be used. Several methods have been proposed [98, 143], among the most famous is the weighted histogram analysis method (WHAM) [96–98]. The WHAM method aims at minimizing the statistical error of the global distribution $P^u(q)$. The global distribution is calculated from the distributions of the single windows as follows

$$P^u(q) = \sum_i^{windows} p_i(q)P_i^u(q) \tag{3.49}$$

The weights p_i are chosen to minimize the statistical error σ such that

$$\frac{d\sigma^2(P^u)}{dp_i} = 0 \tag{3.50}$$

with the condition $\sum_i p_i = 1$. From this it follows

$$p_i = \frac{a_i}{\sum_j a_j}, \quad a_i(q) = N_i e^{-\beta\omega_i(q)+\beta F_i} \tag{3.51}$$

N_i is the total number of steps sampled for window i. The F_i are calculated with

$$e^{-\beta F_i} = \int P^u(q) e^{-\beta \omega_i(q)} dq \tag{3.52}$$

Since P^u enters Equation (3.52) and F_i enters Equation (3.49) via Equation (3.51) the problem has to be solved iteratively.

3.4.2. Free energy perturbation

As the name implies, the FEP method has been developed taking ideas from perturbation theory. Let us assume a N particle system, with Hamiltonian $\mathcal{H}_0(x, p_x)$ with 3N momenta p_x and coordinates x. Let $\mathcal{H}_1(x, p_x)$ be the complete Hamiltonian of the target system defined as

$$\mathcal{H}_1(\mathbf{x}, \mathbf{p}_x) = \mathcal{H}_0(\mathbf{x}, \mathbf{p}_x) + \triangle\mathcal{H}(\mathbf{x}, \mathbf{p}_x) \tag{3.53}$$

We are now interested in the free energy difference between two different states for example the free energy of solvation of a given system at infinite dilution. Taking the relationship between the free energy A and the partition function Q, where Q_b is the partition function of the target and Q_a the partition function of the reference state it follows

$$\triangle A = -\frac{1}{\beta} ln \frac{Q_b}{Q_a} \tag{3.54}$$

and

$$Q(N, V, T) = \frac{1}{h^{3N} N!} \int\int e^{[-\beta \mathcal{H}(q, p_x)]} d\mathbf{x}\ d\mathbf{p}_x \tag{3.55}$$

We can now substitute (3.55) into (3.54) while taking into account equation (3.53) to get

$$\begin{aligned} \triangle A &= -\frac{1}{\beta} ln \frac{\int\int e^{[-\beta \mathcal{H}_1(\mathbf{x}, \mathbf{p}_x)]} d\mathbf{x}\ d\mathbf{p}_x}{\int\int e^{[-\beta \mathcal{H}_0(\mathbf{x}, \mathbf{p}_x)]} d\mathbf{x}\ d\mathbf{p}_x} \\ &= -\frac{1}{\beta} ln \frac{\int\int e^{[-\beta \triangle\mathcal{H}(\mathbf{x}, \mathbf{p}_x)]} e^{[-\beta \mathcal{H}_0(\mathbf{x}, \mathbf{p}_x)]} d\mathbf{x}\ d\mathbf{p}_x}{\int\int e^{[-\beta \mathcal{H}_0(\mathbf{x}, \mathbf{p}_x)]} d\mathbf{x}\ d\mathbf{p}_x} \end{aligned} \tag{3.56}$$

Using the definition of the probability density function which is

$$P_0(\mathbf{x}, \mathbf{p}_x) = \frac{e^{[-\beta \mathcal{H}_0(\mathbf{x}, \mathbf{p}_x)]}}{\int\int e^{[-\beta \mathcal{H}_0(\mathbf{x}, \mathbf{p}_x)]} d\mathbf{x}\ d\mathbf{p}_x} \tag{3.57}$$

Equation (3.54) becomes

$$\begin{aligned} \triangle A &= -\frac{1}{\beta}\int\int e^{[-\beta\triangle\mathcal{H}(\mathbf{x},\mathbf{p}_x)]}P_0(\mathbf{x},\mathbf{p}_x)d\mathbf{x}\ d\mathbf{p}_x \\ &= -\frac{1}{\beta}ln\left\langle e^{[-\beta\triangle\mathcal{H}(\mathbf{x},\mathbf{p}_x)]}\right\rangle_0 \end{aligned} \tag{3.58}$$

which is the basic FEP formula. The ensemble average of sampled configurations of the reference state is denoted by $\langle\ldots\rangle_0$. Hence it is possible to estimate $\triangle A$ by sampling just the reference state. Integration of the kinetic part can be done analytically and cancels out in Equation (3.54).Therefore, the above expression simplifies to

$$\triangle A = -\frac{1}{\beta}ln\left\langle e^{-\beta\triangle U}\right\rangle_0 \tag{3.59}$$

Here U is the potential energy. The FEP method is based on this last equation. The basic implementation is discussed in Chapter 4.

4. Implementation

All following algorithms have been implemented into the Conformational Analysis and Search Tool (CAST) [144] developed in the working group of Engels. The program features several force field implementations as well as interfaces to other programs. Force field implementations include CHARMM [145,146], AMBER [147], OPLS-AA [148,149], AMOEBA [129] and a newly developed SAPT force field [150, 151] implementation which is based on physically grounded electrostatics. For semi-empirical calculations an interface to MOPAC [152] can be used. In this work the possibilities of CAST have been further extended. Treatment of electrostatics has been improved by implementation of a smooth particle mesh Ewald (SPME) method for the standard force fields OPLS-AA, CHARMM and AMBER. The performance has been increased by a partial parallelization of the inherent force fields (OPLS-AA, CHARMM, AMBER). Furthermore, a linked cell algorithm has been implemented to improve the efficiency of building up the Verlet list. For the simulation of dynamic properties a molecular dynamics code has been included, featuring several algorithms for integration, temperature and pressure control as well as boundary potentials and constraints. Based upon this MD code umbrella sampling and free energy perturbation techniques have been included. For DFT calculations a MPI interface to the GPU accelerated program TeraChem is now available. In the following chapter the theory and implementation of these algorithms are discussed in more detail.

4.1. Performance

4.1.1. Linked cell algorithm

One of the most time consuming tasks in force field calculations is the calculation of pairwise interactions, namely van-der-Waals and charge interactions. The most straightforward algorithm to compute pairwise interactions within a given cutoff distance relies on a double loop over all unique atom pairs in the reference box, leading to a scaling of the computational cost as $\mathcal{O}[N(N-1)/2] \approx \mathcal{O}[N^2]$, where N is the number of atoms in the system. Computational cost can be reduced by the use of a cutoff radius, at which

the interactions are truncated, in combination with a Verlet list [49]. The pairwise interactions are evaluated using two steps. The first is the generation of the Verlet list and the second the evaluation of the pairwise interactions. Updating the pairlist only every x steps (depending on the implementation this can be up to 50 steps) further saves computational time. Using a Verlet list based on a cutoff radius has some drawbacks, the loss of accuracy if interactions are truncated and the fact that the Verlet list is constant between updates. One method to decrease the loss of accuracy is to use an extended Verlet list, where the pairlist is generated with a slightly bigger radius than the cutoff radius [49]. Another method relies on non regular update intervals, where updates are performed if the net movement of atoms exceeds a certain value. Generally two main strategies exist for the speedup of pairwise interaction computations. The first is the acceleration of the energy, gradient and virial computation for example by parallelization (see Section 4.1.2), the second the development of faster algorithms for pairlist generation, at best with linear scaling. Linear scaling algorithms depend on the partitioning of the simulation box in smaller units or sub cells. These subcells normally are of the size of the cutoff radius. Algorithms based on this kind of subdivison are called linked-cell or linked-list algorithms [153–155]. The basic idea is that all possible interaction partners of a given atom can be found in its respective subcell or the 26 neighboring cells. If the total simulation box is big enough compared to the subcell length, this algorithm scales linearly.

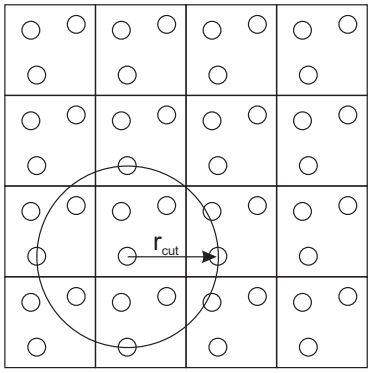


Figure 4.1.: Two-dimensional picture of a linked cell algorithm. The whole system is divided into subcells with a side length equal to the cutoff radius. The cutoff radius is shown as a circle. Only atoms which reside in the neighbouring cells of the current atom are possible interaction partners.

In a first step the total size of the system in each dimension is calculated. A cubic box is set around and is divided into small subcells. Each subcell is then assigned a cellvector (*cellx, celly, cellz*) and a cell number (*ncell*). After the cells have been generated and

labeled the atoms are sorted into the cells and the information is stored in two arrays. The HEAD array is of the size *ncell* and the LIST array has the size of number-of-atoms (*natom*). The loop runs over all atoms and the sorting is done recursively. For each atom the cellvector and cellindex are calculated. When this is done the position i (i = index number of the atom) in the LIST array gets assigned the value at position *ncell* in the HEAD array (with *ncell* = cellnumber of the current examined atom). After that the position *ncell* in HEAD gets assigned the index number i of the currently examined atom. This pattern goes on over all atoms. In the end the atoms have been sorted in reverse into the two arrays. The sorting is shown in Algorithm 1 as pseudocode.

Algorithm 1 Code for the recursive sorting of the atoms into the two arrays HEAD and LIST using a linked cell algorithm

```
1: for i=1; i < natoms do
2:     compute cell number ncell of current atom i;
3:     LIST[i] = HEAD[ncell];
4:     HEAD[ncell] = i;
5: end for
```

The values of the HEAD array position indicate the index number of the first atom found in its respective cell. The current number of the atom found in the cell acts furthermore as a pointer to the next atom in the cell. After the sorting, the two arrays can be translated into the Verlet list. Figure 4.2 shows the HEAD and LIST arrays after sorting of the atoms is finished. The pointers are indicted by arrows.

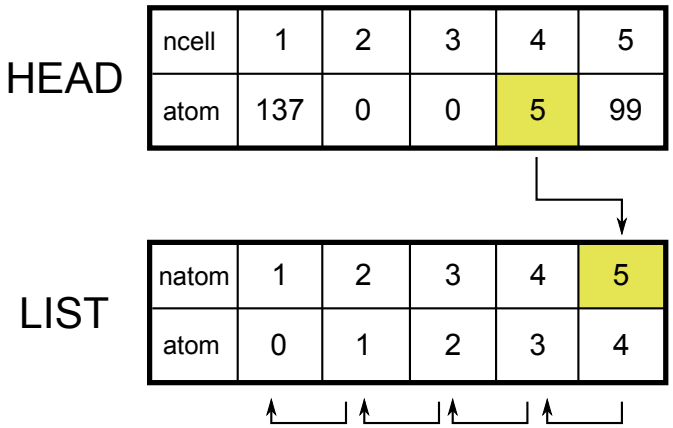


Figure 4.2.: Overview over the HEAD and LIST arrays. HEAD contains the atom with the highest index number in each cell. LIST contains all atoms. The index number serves as a pointer to the next atom found in the respective cell.

Algorithm 2 Turning the linked cell list into a Verlet list

```
1: for i=1; i < natoms do
2:     compute cell number ncell of current atom i;
3:     for all neighbouring cells do
4:         get first atom in cell: atom j = HEAD[ncell];
5:         while j ≠ 0 do
6:             r^2 = |r_i − r_j|^2
7:             if r^2 ≤ r_c^2 then
8:                 store interaction i, j
9:             end if
10:            j = LIST[j];
11:        end while
12:    end for
13: end for
```

The algorithm for the translation of the linked-list to a verlet list is shown in Algorithm 2. For each atom the cell number is calculated, and only the atoms in the current and neighboring cells are evaluated using the HEAD and LIST arrays. For the first neighboring cell the atom with the highest index number in the cell is calculated. With the index of this atom the LIST array is used to loop over the remaining atoms in a cell until index 0 is reached. This is repeated for all other neighboring cells. The main computational time during list build up is produced by the calculation of the distance between two possible interaction partners. The use of the cell decomposition dramatically reduces the distance calculations since many of the atoms for which the calculation is spurious are already sorted out.

4.1.2. Parallelization

Parallelization has been increasingly successful with the development of multi-core CPU's. Depending on the architecture of the underlying computing machine and especially the memory type, several different parallelization models are available. The memory types can basically be divided into two main categories, distributed and shared memory. In distributed memory, the single nodes are coupled by a node interconnect, each node has a fast access to its memory, but a slow access to the other nodes memory. This is also called a non-uniform memory access or NUMA [156]. Shared memory architectures use the opposite possibility. Here, all CPUs are connected to all memory banks with the same speed, which leads to a uniform memory access (UMA) [156],

also called symmetric multi processing (SMP) [156]. The parallel routines of CAST were developed with a UMA architecture in mind. For this, the OpenMP [157] library was used. OpenMP uses shared memory directives and has the great advantage that synchronization between threads is implicit, meaning the user can, but doesn't have to, take care of synchronization manually. Based on a fork-join model, see Figure 4.3, OpenMP is especially suited for loop parallelization.

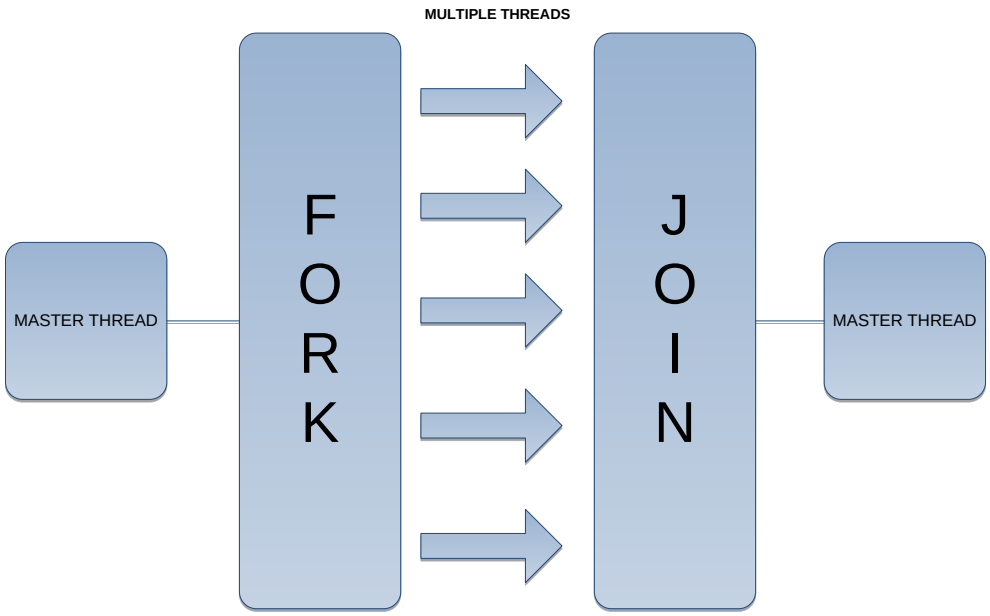
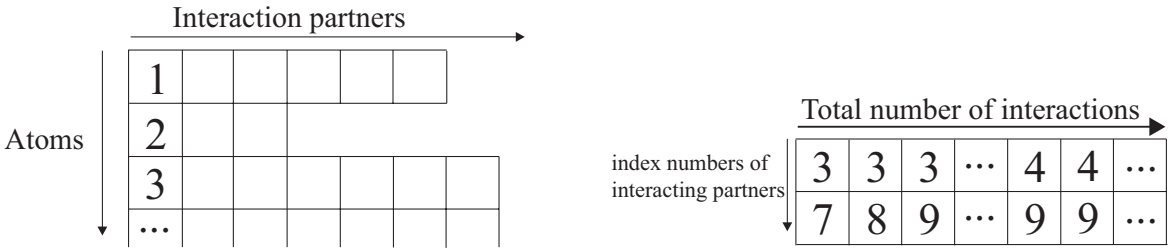


Figure 4.3.: Fork-join model of the OpenMP directive. The master thread executes the program in serial until a parallel region is reached (fork). The work is distributed among a number of threads and executed in parallel. When the end of the parallel region is reached, the master thread continues the program execution.

As mentioned in Section 5.1.2 the most time consuming function is the non-bonded potential- and gradient evaluation, which makes it the ideal target for parallelization. If a cutoff radius is used for the non-bonded interactions, van-der-Waals and electrostatic interactions are calculated for the same atom pairs and are processed in the same function. The parallel implementation now depends only on the way the interaction pairs are stored. This list can be constructed in two ways. The first possibility is a two-dimensional array with all atoms on one axis and the interacting atoms on the respective other axis. The alternative is a one-dimensional array, where all interaction pairs are stored in a single array box consisting of the two index numbers of the interacting atoms. The two array types are depicted in Figure 4.4. For a two dimensional array a

double loop is needed. The first loop over all atoms i, the second loop over all atoms j interacting with atom i. Since not all atoms have the same number of interaction partners the size of the second array dimension is different for each atom. Therefore, a static work distribution between threads at the start of the outer loop can lead to massive overhead if for example thread one happens to calculate only half the number of interactions than thread two. To avoid this problem OpenMP provides dynamic load balancing. At the beginning of the outer loop each threads get assigned a static number of loop iterations. When the thread finishes it instantly gets the next iterations until the loop is finished which guarantees a smooth work distribution among threads. A disadvantage is the overhead produced by assigning the workload dynamically during loop execution. This problem is avoided if a one-dimensional array is used. Since each array box contains the exact same number of interactions, namely one, the work has only to be distributed once at the beginning of the loop. It is equally divided among all threads and the overhead is minimal.



(a) Schematic of a Verlet list for dynamic allocation

(b) Schematic of a Verlet list for static allocation

Figure 4.4.: Verlet non-bonded list examples. List **a** is a two dimensional vector with the atom label on one side and the interacting partners in the other dimension. List **b** is a one dimensional vector with a single interaction pair at each entry.

Parallelization has the inherent problem that sometimes more than one process tries to calculate values that are stored in a single variable (e.g. energy). Uncontrolled access of more than one thread to a specific memory block is called a data race. Data races lead to uncontrolled program behavior and wrong results. For this case OpenMP provides an implicit solution, the definition of private variables and the reduction clause. If a variable is declared private, a copy, specific for each executing thread, is created for each private variable and each thread accumulates the results in its private copy. Another addition to the OpenMP library is the reduction clause. If a variable is included in the reduction clause, the values of the private copies are accumulated at the end of the parallel region automatically to give the final result (see Figure 4.5).

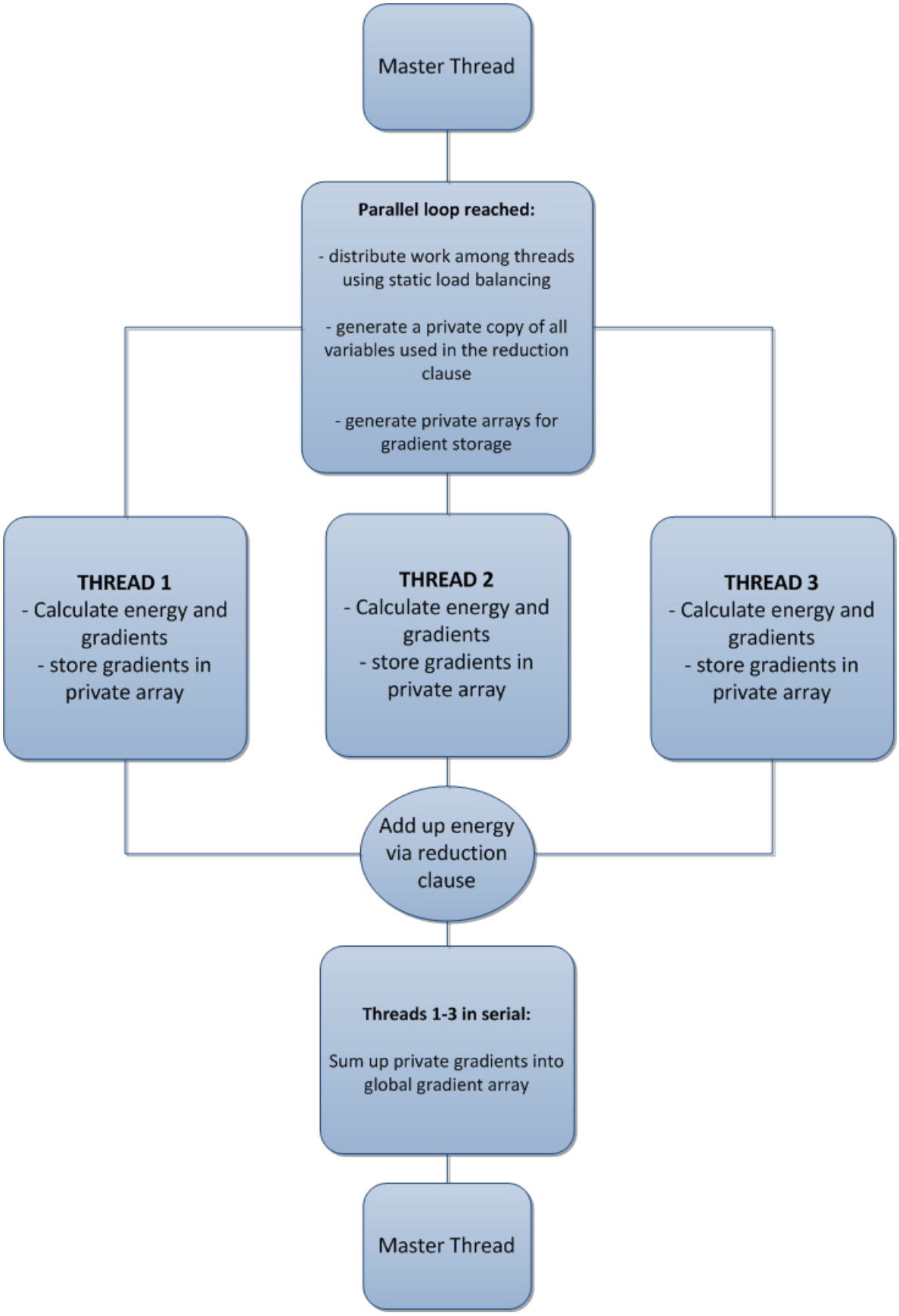


Figure 4.5.: Flowchart of the CAST OpenMP implementation.

During the execution of the non-bonded interaction calculations, this is used to evaluate the non-bonded energy. Unfortunately this is not possible for arrays. For each pairwise interaction, 3 gradient values are calculated. These need to be stored in private copies of the gradient array for each thread. At the end of the loop these arrays have to be added up into the global gradient array manually. The parallel workflow is shown in Figure 4.5. The non-bonded implementation in CAST is based on a Verlet list for static allocation (see Figure 4.4 (b)). Performance of the parallelization is discussed in Chapter 5.

4.2. Smooth Particle Mesh Ewald method

Typical values for a cutoff radius in force field computations are 9-12 Å. For the van-der-Waals interactions this cutoff is a reasonable method to speed up computations because at a value of 10Å the strength of a van-der-Waals interaction has been diminished by about 90%. Unfortunately this is not true for electrostatic interactions which can range up to 100Å [122]. Using a cutoff radius can therefore severely alter the results of a simulation. Furthermore, electrostatic interactions are extremely important for a lot of reactions and interactions, especially in biomolecular systems. If a finite system is used, the standard Coulomb interaction can be calculated for all interactions in the system. If the system is periodic, this is no longer possible because the system is subjected to the minimum image criterion which states that the cutoff radius used has to be smaller or equal half of the periodic box length. In this case one has to refer to methods which allow the computation of full electrostatics for periodic systems. In the early 20th century, Ewald proposed such a method [101, 158], originally designed for the calculation of interactions in crystals.
We consider a system of N particles, whose charges q_i are expressed by Coulomb point charges at the positions $\mathbf{r}_i$ in the unit cell U. The simulation box, the particles reside in has a net charge of zero, a boxlength L and a volume $V = L^3$. The total electrostatic energy under periodic boundary conditions can be written as

$$E(\mathbf{r}_1, ...\mathbf{r}_n) = \frac{1}{2} \sum_{n}^{*} \sum_{i} \sum_{j} \frac{q_i q_j}{|\mathbf{r}_{ij} + n|} \tag{4.1}$$

where $\mathbf{r}_{ij}$ is the distance between particles i and j. Boundary conditions are taken care of by the sum over n and the prime means that in the case of $i = j$ the term $n = 0$ must be omitted. The problem with this summation is that it is only conditionally convergent, meaning that the order of summation has to be precisely defined and that

the long range part of the Coulomb interaction is only very slowly convergent which makes it impractical. A better convergent formula is obtained if $1/r$ is separated into two parts, here using an arbitrary function $f(r)$

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1 - f(r)}{r} \tag{4.2}$$

The basic idea for a practical solution is to split the Coulomb potential into a slowly decaying part at large r, which converges quickly in Fourier space, and a fast varying part at small r which can be calculated in real space [101]. This can be realized by a suitable choice of f. The first part of the summation should be negligible or even vanishing at a certain distance or cutoff radius. In that case the summation from zero to the cutoff distance is a good approximation for the short range contribution to the total electrostatic energy. The second part is supposed to be slowly varying for all r. Since these conditions are not too restrictive several possibilities exist for f. The traditional function used for f is the complementary error function $\mathrm{erf}(r) = 2/\sqrt{\pi}\int_r^\infty \exp(-t^2)dt$. Inserting this function yields the original formula derived by Ewald

$$E = E_{dir} + E_{rec} + E_{self} + E_{dipole} \tag{4.3}$$

with E_{dir} as the real space and E_{rec} as the reciprocal space contribution. E_{self} denotes a self correction term and E_{dipole} a dipole correction. Several years ago, this Ewald summation was adopted for the use in molecular simulations subjected to periodic boundary conditions. This particle mesh Ewald (PME) [159] method is based on the original method proposed by Ewald, but uses a grid to extrapolate the charges and speed up the computation. Several different implementations have been proposed over the last years. In CAST the smooth particle mesh Ewald (SPME) [160] method proposed by Darden and Pedersen has been implemented. In contrast to the original implementation [161] which used Lagrangian extrapolation, the SPME method uses cardinal B-splines. This has the advantage that an analytical expression for the atomic forces can be obtained. In the following the SPME method and general CAST implementation is described.

Consider a system consisting of N point charges $q_1, q_2, ...q_n$ in a unit cell U. The positions of the point charges are given by $\mathbf{r}_1, \mathbf{r}_2, ...\mathbf{r}_n$ and the vectors $\mathbf{a}_\alpha, \alpha = 1, 2, 3$ represent the orthogonal lattice vectors. Accordingly the reciprocal lattice vectors $\mathbf{a}^*_\alpha$ are defined by $\mathbf{a}^*_\alpha \cdot \mathbf{a}_\beta = \delta_{\alpha\beta}$ for $\alpha, \beta = 1, 2, 3$. Each point charge can be assigned a fractional coordinate $s_{\alpha i}, \alpha = 1, 2, 3$ defined as $s_{\alpha i} = \mathbf{a}^*_\alpha \cdot \mathbf{r}_i$. The interaction of the charges is based on Coulombs law under periodic boundary conditions, meaning that

a point charge q_i interacts with all other charges including their and its own periodic images at the positions $\mathbf{r}_i + n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3$, for all n_1, n_2, n_3 integer values. Thus the electrostatic energy of the unit cell reads

$$E(\mathbf{r}_1, ...\mathbf{r}_n) = \frac{1}{2}\sum_{\mathbf{n}}^{*}\sum_{i}\sum_{j}\frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}|} \tag{4.4}$$

where the outer sum is over all vectors $\mathbf{n}$ ($\mathbf{n} = n_1\mathbf{a} + n_2\mathbf{a} + n_3\mathbf{a}$). Terms where $\mathbf{n} = 0$ and $i = j$ are neglected as indicated by the prime. In most force fields the potential functions governing the non-bonded interactions have been parametrized in a way that some interactions have to be neglected. Only interactions starting from 1-4 interactions are calculated (1-2 and 1-3 covalent interactions are omitted). Unfortunately these interactions are intrinsically included in Equation (4.4). Therefore, a correction term must be applied to the direct and reciprocal sum in the calculation. The electrostatic energy can then be calculated according to Equation (4.3) but without the dipole term. The potentials are defined as follows

$$E_{dir} = \frac{1}{2}\sum_{n}^{*}\sum_{i,j=1}^{N}\frac{q_i q_j erfc(\beta|\mathbf{r}_j - \mathbf{r}_i + \mathbf{n}|)}{|\mathbf{r}_j - \mathbf{r}_i + \mathbf{n}|} \tag{4.5}$$

$$E_{rec} = \frac{1}{2\pi V}\sum_{\mathbf{m}\neq 0}\frac{exp(-\pi^2\mathbf{m}^2/\beta^2)}{\mathbf{m}^2}S(\mathbf{m})S(-\mathbf{m}) \tag{4.6}$$

$$E_{self} = -\frac{1}{2}\sum_{(i,j)\in M}\frac{q_i q_j erf(\beta|\mathbf{r}_i - \mathbf{r}_j|)}{|\mathbf{r}_i - \mathbf{r}_j|} - \frac{\beta}{\sqrt{\pi}}\sum_{i=1}^{N} q_i^2 \tag{4.7}$$

with *erfc* as the complementary error function, β as the Ewald coefficient, V the volume and $\mathbf{m} = (m_1\mathbf{a}_1^*, m_2\mathbf{a}_2^*, m_3\mathbf{a}_3^*)$ as the reciprocal lattice vectors. The asterisk again means that terms with $\mathbf{n} = 0$ and $i = j$ are neglected in Equation (4.5). The factor $S(\mathbf{m})$ is called the structure factor and given by

$$\begin{aligned} S(\mathbf{m}) &= \sum_{j}^{N} q_j exp(2\pi i\ \mathbf{m}\cdot\mathbf{r}_j) \\ &= \sum_{j}^{N} q_j exp\left[2\pi i(m_1 s_{1j} + m_2 s_{2j} + m_3 s_{3j}\right] \end{aligned} \tag{4.8}$$

Here $s_{\alpha j}, \alpha = 1, 2, 3$ are the fractional coordinates of atom j. The approximate structure factors can now be obtained by spline interpolation of the complex exponential $\exp(2\pi i\mathbf{m}\cdot\mathbf{r})$. Using the positive integers K_1, K_2 and K_3 the scaled fractional coordinates u_1, u_2, u_3 are $u_\alpha = K_\alpha a_\alpha^*\cdot\mathbf{r}$ for $\alpha = 1, 2, 3$. With this, the exponential expression

becomes

$$exp(2\pi i\ \mathbf{m}\cdot\mathbf{r}) = exp\left(2\pi i\frac{m_1}{K_1}u_1\right)\cdot exp\left(2\pi i\frac{m_2}{K_2}u_2\right)\cdot exp\left(2\pi i\frac{m_3}{K_3}u_3\right) \tag{4.9}$$

The interpolation is now done with cardinal B-splines $M_{2p}(u)$. The linear hat function $M_2(u)$ is given by $M_2(u) = 1 - |u-1|$ for $0 \leq u \leq 2$ and $M_2(u) = 0$ for $u < 0$ or $u > 2$. For any n greater than 2 the spline is defined by the following recursive formula

$$M_n(u) = \frac{u}{n-1}M_{n-1}(u) + \frac{n-u}{n-1}M_{n-1}(u-1) \tag{4.10}$$

Using spline interpolation for the exponential factor, the final expression reads

$$exp\left(2\pi i\frac{m_i}{K_i}u_i\right) \approx b_i(m_i)\sum_{k=-\infty}^{\infty} M_n(u_i-k)\cdot exp\left(2\pi i\frac{m_i}{K_i}k_i\right) \tag{4.11}$$

where $b_i(m_i)$ are Euler exponential splines of the form

$$b_i(m_i) = exp\left(2\pi i(n-1)m_i/K_i\right)\times\left[\sum_{k=0}^{n-2} M_n(k+1)exp(2\pi i m_i k/K_i\right]^{-1} \tag{4.12}$$

These exponential splines are independent of the particle coordinates and only have to be computed once at the beginning of the simulation. The final expression for the approximate structure factor now reads

$$\widetilde{S}(\mathbf{m}) = b_1(m_1)b_2(m_2)b_3(m_3)F(Q)(m_1,m_2,m_3) \tag{4.13}$$

where $F(Q)$ is the discrete Fourier transform of the charge grid Q which has the dimensions $K_1 \times K_2 \times K_3$ and is given by

$$\begin{aligned} Q(k_1,k_2,k_3) = &\sum_{i=0}^{N}\sum_{n_1,n_2,n_3} q_i M_n\left(u_{1i}-k_1-n_1K_1\right) \\ &\times M_n\left(u_{2i}-k_2-n_2K_2\right) \\ &\times M_n\left(u_{3i}-k_3-n_3K_3\right) \end{aligned} \tag{4.14}$$

The reciprocal space energy can now be approximated by

$$\begin{aligned}\widetilde{E}_{rec} &= \frac{1}{2\pi V}\sum_{m\neq 0}\frac{exp\left(-\pi^2\mathbf{m}^2/\beta^2\right)}{\mathbf{m}^2}B(m_1,m_2,m_3) \\ &\cdot F(Q)(m_1,m_2,m_3)F(Q)(-m_1,-m_2,-m_3) \\ &= \frac{1}{2}\sum_{m_1=0}^{K_1-1}\sum_{m_2=0}^{K_2-1}\sum_{m_3=0}^{K_3-1}Q(m_1,m_2,m_3) \\ &\cdot (\Phi_{rec}\star Q)(m_1,m_2,m_3)\end{aligned} \tag{4.15}$$

with the pair potential $\Phi_{rec} = F(B\cdot C)$. The arrays B and C are given by the following expressions

$$B(m_1,m_2,m_3) = \prod_{i=1}^{3}\left|b_i(m_i)\right|^2 \tag{4.16}$$

$$C(m_1,m_2,m_3) = \frac{1}{\pi V}\frac{exp(-\pi^2\mathbf{m}^2/\beta^2)}{\mathbf{m}^2} \tag{4.17}$$

From this expression the particle forces can be derived by differentiating with respect to $\mathbf{r}_i$. The atomic forces are given by

$$\begin{aligned}\frac{\partial\widetilde{E}_{rec}}{\partial\mathbf{r}_{\alpha i}} &= \sum_{m_1=0}^{K_1-1}\sum_{m_2=0}^{K_2-1}\sum_{m_3=0}^{K_3-1}\frac{\partial Q}{\partial\mathbf{r}_{\alpha i}}(m_1,m_2,m_3) \\ &\cdot (\Phi_{rec}*Q)(m_1,m_2,m_3)\end{aligned} \tag{4.18}$$

During program execution, the following steps are performed. Before the simulation is started the Ewald coefficient β, see Equations (4.5), (4.7), is calculated. CAST uses an initial guess based on the chosen cutoff radius and a trial and error function. This result is further refined by a binary search method. Next the grid is set up in a way that the grid size in each dimension is an even number and consists of prime factors of 2, 3 and 5 only. This ensures a maximum performance of the FFTW algorithm [162] which is used during the energy calculation. Then the arrays B and C are assigned and BC is computed and stored in memory. When the simulation is started, the following calculations have to be carried out at each step:

1. calculate the fractional coordinates and fill the array Q.

2. calculate the Fourier transformation of the charge array Q and overwrite itself with the Fourier transform result.

3. use the Fourier transformed array Q to calculate the reciprocal energy according to Equation (4.15). At the same time compute the virial components and overwrite the array Q again with the product of itself and the array BC.

4. Perform an inverse Fourier transform of the resulting array to arrive at the convolution $(\Phi_{rec} \star Q)$.

5. Compute the per particle forces according to Equation (4.18).

To arrive at the total electrostatic energy the reciprocal space part is now added to the direct and correction part according to Equations (4.5) - (4.7). The correction term has to be split into two parts. The first part accounts for the self-interaction of each atom and is calculated using a single loop over all atoms. The second part accounts for the 1-2 and 1-3 non-covalent interactions which are normally neglected in a force field calculation. The direct space part uses already existing functions within CAST (bond and angle energy calculation functions) which are slightly modified to match energy calculation according to Equation (4.5). For the reciprocal space part the Fourier transformation is performed with version 3.3.4 of the FFTW package [163]. The general procedure for a calculation using SPME can be seen in Figure 4.6. Most of the PME parameters are fixed values but two parameters can be adjusted. The first one is the order of the B-Splines which can be set to a maximum of 10. The standard B-spline order is 5 which is sufficient for almost any calculation. The second parameter is the tolerance for the calculation of the Ewald parameter which affects the overall accuracy. The standard value is 1e-8.

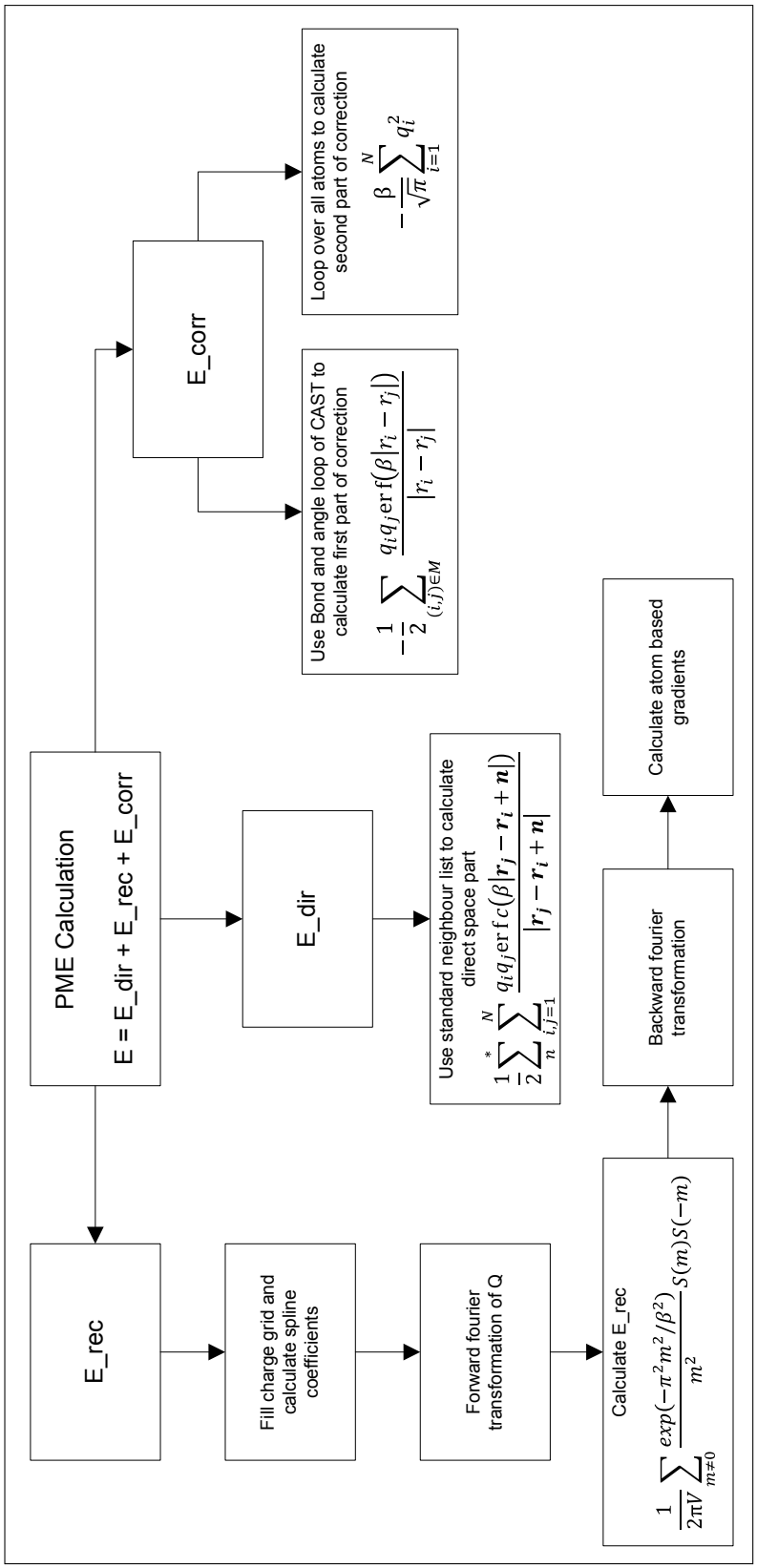


Figure 4.6.: Flowchart for the calculation of the electrostatic contribution using a smooth particle mesh Ewald method. The electrostatic correction and direct sum energies can be calculated using existing code in the CAST program. The reciprocal energies and gradients make use of the fast Fourier transform package.

4.3. Molecular Dynamics

The molecular dynamics implementation features several algorithms which make it suitable for simulations in the NVE, NVT and NPT ensembles (see Section 1.3). Two different integration schemes are available, the velocity Verlet [52] and a modified version of the Beeman [164, 165] integrator. The temperature can be controlled by means of direct velocity scaling and a Nòse-Hoover thermostat. Pressure changes are realized via a Berendsen coupling scheme. Specific to the velocity Verlet integrator, constraints via the RATTLE [166, 167] algorithm can be applied to bond lengths. For the simulation of cluster systems spherical boundary conditions are available, for infinite systems periodic boundary conditions have been implemented. The MD code can also be used in conjunction with the CAST interfaces to MOPAC [152] and TeraChem [168, 169]. In the following chapter, the implementations of the different methods are discussed in detail.

4.3.1. General features

The MD part of CAST has been implemented as an own task in the program. Upon initialization of the MD subroutine several standard procedures are executed regardless of the user input. The standard procedures can be seen in Table 4.1.

Standard initialization	
1.	Compute total molecular mass
2.	Assign initial velocities
3.	Calculate degrees of freedom
4.	Calculate geometric center and center of mass
5.	Remove initial angular and translational momentum

Table 4.1.: Overview over the standard routines which are initialized before every molecular dynamics run.

The initial velocities depend on the temperature input of the user. If no temperature is set at all or the temperature is set to zero, all initial velocities will be zero. If the initial temperature is not zero the velocities are taken from a Maxwell-Boltzmann distribution with the temperature set by the user. The degrees of freedom are calculated taking into account possible constraints set for the simulation via a constraint algorithm (see Section 4.3.4). They are needed for the calculation of the instantaneous temperature. When velocities are set randomly from a distribution, the system will have an angular

and translational momentum. To avoid a drift of the system the initial translational motion is removed from the particles by substracting the translational contribution from the velocities. The same is done for the angular part of the velocities to avoid rotation of the system.

4.3.2. Integrators

The main part of the MD code are the integrators which employ Newton's equation of motion to progress the system in time. One of the most common integrators is the velocity Verlet integrator. A more sophisticated method is Beeman's integration scheme which is a predictor - corrector method.

Velocity Verlet integrator [52] The velocity Verlet integrator is an improved version of the original Verlet integration scheme in which the velocities of the particles were not explicitly available. It stores acceleration, velocities and positions at the same time. For a simulation containing no further constraints or boundary conditions in the NVE ensemble, the psueod-code is given in the following Algorithm 3

Algorithm 3 MD simulation using the velocity Verlet integrator

1: **procedure** VELOCITY VERLET INTEGRATION
2: **for** each integration step **do**
3: $r_{n+1} = r_n + v_n \triangle t + \frac{1}{2}\left(\frac{F_n}{m}\right)\triangle t^2$
4: $v_{n+1/2} = v_n + \frac{1}{2}\left(\frac{F_n}{m}\right)\triangle t$
5: **Calculate new atomic forces**
6: $v_{n+1} = v_{n+1/2} + \frac{1}{2}\left(\frac{F_{n+1}}{m}\right)\triangle t$
7: **end for**
8: **end procedure**

The application of the integrator consists of two steps. First, the full step progression of the positions and the half step progression of the velocities is done. Second the calculation of new gradients based on the new positions, and the progression of the half step velocities to the full step velocities is performed. If constraints or advanced simulation techniques are used the algorithm is enhanced to

Algorithm 4 MD simulation using velocity Verlet in advanced ensembles

procedure VELOCITY VERLET INTEGRATION
  for each integration step **do**
    apply half step temperature and pressure corrections
    $r_{n+1} = r_n + v_n \triangle t + \frac{1}{2}\left(\frac{F_n}{m}\right)\triangle t^2$
    $v_{n+1/2} = v_n + \frac{1}{2}\left(\frac{F_n}{m}\right)\triangle t$
    apply half step constraint corrections
    calculate new atomic forces
    apply spherical boundary corrections
    $v_{n+1} = v_{n+1/2} + \frac{1}{2}\left(\frac{F_{n+1}}{m}\right)\triangle t$
    apply full step constraint corrections
    apply full step temperature and pressure corrections
  end for
end procedure

Beeman integrator As a second integration method for the equations of motion the Beeman algorithm is implemented. Beeman's algorithm is a method for the integration of second order ordinary differential equations. In his publication Beeman proposed different predictor-corrector methods [164, 165]. These were adapted to the equations of motion. The most popular of these methods is an order 3 method which is related to the velocity Verlet integration scheme but is more accurate in velocities. The positions and velocities disturbed by $\triangle t$ are calculated by

$$x(t + \triangle t) = x(t) + v(t)\triangle t + \frac{2}{3}\left(\frac{F_{t+\triangle t}}{m}\right)\triangle t - \frac{1}{6}\left(\frac{F_{t-\triangle t}}{m}\right) \tag{4.19}$$

$$v(t + \Delta t) = v(t) + \frac{1}{3}\left(\frac{F_{t+\Delta t}}{m}\right)\Delta t + \frac{5}{6}\left(\frac{F_t}{m}\right)\Delta t - \frac{1}{6}\left(\frac{F_{t-\Delta t}}{m}\right)\Delta t \tag{4.20}$$

The basic algorithm for a MD simulation using Beemans algorithm is almost identical to the velocity Verlet. First the new positions and the half step velocities are calculated. After that new atomic forces are generated and the velocities are updated to the new time step. An algorithm for the Beeman integration using advanced sampling techniques can be seen in the following Algorithm 5

Algorithm 5 MD simulation using Beeman integration in advanced ensembles

1: **procedure** BEEMAN INTEGRATION
2: **for** each integration step **do**
3: apply half step temperature and pressure corrections
4: $v(t+\frac{1}{2}\Delta t) = v(t) + \frac{2}{3}\left(\frac{F_t}{m}\right)\Delta t - \frac{1}{6}\left(\frac{F_{t-\Delta t}}{m}\right)\Delta t$
5: $x(t+\Delta t) = x(t) + v(t+\frac{1}{2}\Delta t)\cdot\Delta t$
6: apply half step constraint corrections
7: **calculate new atomic forces**
8: apply spherical boundary corrections
9: $v(t+\Delta t) = v(t+\frac{1}{2}\Delta t) + \frac{1}{3}\left(\frac{F_{t+\Delta t}}{m}\right)\Delta t + \frac{1}{6}\left(\frac{F_t}{m}\right)\Delta t$
10: apply full step constraint corrections
11: apply full step temperature and pressure corrections
12: **end for**
13: **end procedure**

4.3.3. Temperature and Pressure

Temperature control in MD simulations is one of the most important features since it allows the sampling in the canonical ensemble. Several different possibilities exist to control the temperature, all of them based on the scaling of particle velocities.

Temperature by direct velocity scaling The simplest way to control the instantaneous temperature is by directly scaling the velocities of each particle. The scaling factor λ is determined by the following formula

$$\lambda = \sqrt{\frac{T_0}{T}} \tag{4.21}$$

with T_0 as desired and T as current temperature. Direct scaling of the velocities is a very drastic and unphysical, yet efficient way to match the exact desired temperature. Since it doesn't reproduce any desired ensemble the direct velocity scaling is usually only used during the heating phase of the calculation. CAST automatically turns to a thermostat after heating is done and the temperature is supposed to stay constant.

Thermostat methods In order to reproduce a canonical ensemble the Nóse-Hoover thermostat [58, 59] has been implemented into CAST. The implementation follows the decomposition scheme of Martyna [170] and the algorithm derived in the book of Frenkel and Smit [4]. The general idea is to add an additional coordinate s to the system.

Simulating this extended system in an NVE ensemble generates a canonical ensemble in the real system. The equations of motion for the extended system read

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} \tag{4.22}$$

$$\dot{\mathbf{p}}_i = -\frac{\partial U}{\partial \mathbf{r}_i} - \zeta \mathbf{p}_i \tag{4.23}$$

$$\dot{\zeta} = \left(\sum_i \left(\frac{p_i^2}{m_i} \right) - \frac{L}{\beta} \right) Q^{-1} \tag{4.24}$$

$$\frac{\dot{s}}{s} = \frac{d\ ln\ s}{dt} = \zeta \tag{4.25}$$

Here $\mathbf{p}$ denotes the momenta and m the mass of the particle i. Q is an effective mass associated to s, L a fixed parameter and ζ is the thermodynamic friction coefficient. In order to generate a time-reversible algorithm for the numerical solution of these equations we have to use a Liouville equation combined with a Trotter expansion. For the Nóse-Hoover system the Liouville operator can be defined as

$$iL_{NHC} = iL_r + iL_v + iL_C \tag{4.26}$$

with iL_r as the part which only involves positions, iL_v the part which only involves velocities and iL_C the part for the Nóse-Hoover thermostat. The single components of the operator are further defined as

$$iL_r = \sum_{i=1}^{N} \mathbf{v}_i \cdot \nabla_{\mathbf{r}_i} \tag{4.27}$$

$$iL_\nu = \sum_{i=1}^{N} \frac{\mathbf{F}_i(\mathbf{r}_i)}{m_i} \cdot \nabla_{\mathbf{v}_i} \tag{4.28}$$

$$iL_C = \sum_{k=1}^{M} \nu_{\zeta_k} \frac{\partial}{\partial \zeta_k} - \sum_{i=1}^{N} \nu_{\zeta_1} \mathbf{v}_i \cdot \nabla_{\mathbf{v}_i} \tag{4.29}$$

$$+ \sum_{k=1}^{M-1} \left(G_k - \nu_{\zeta_k} \nu_{\zeta_{k+1}} \right) \frac{\partial}{\partial \nu_{\zeta_k}} + G_M \frac{\partial}{\partial \nu_{\zeta_M}} \tag{4.30}$$

The Liouville operator can now be factorized using the Trotter expansion. A detailed description of the procedure is given in reference [170]. Here only the results for the implementation as a numerical algorithm are given. Factorization of the Liouville operator

yields

$$e^{iL\Delta t} = e^{(iL_C\Delta t/2)}e^{(iL_v\Delta t/2)}e^{(iL_r\Delta t/2)}e^{(iL_v\Delta t/2)}e^{(iL_C\Delta t/2)} + \mathcal{O}(\triangle t^3) \tag{4.31}$$

The part for the Nóse-Hoover chain iL_C can be further factorized. In this example we will use a chain length of $M = 2$. In [170] the general case is discussed. For a chain length of 2 the Nóse-Hoover part of the Liouville operator yields five terms, derived from equation (4.30)

$$iL_C = iL_\zeta + iL_{C_\nu} + iL_{G1} + iL_{\nu\zeta_1} + iL_{G2} \tag{4.32}$$

Further decomposing the iL_C part with a Trotter expansion yields (again only the result is shown)

$$\begin{aligned} e^{iL_C\Delta t/2} =& e^{iL_{G2}\Delta t/4}\left[e^{iL_{v\zeta 1}\Delta t/8}e^{iL_{G1}\Delta t/4}e^{iL_{v\zeta 1}\Delta t/8}\right] \\ &\times e^{iL_\zeta\Delta t/2}e^{iL_{C\nu}\Delta t/2} \\ &\times \left[e^{iL_{v\zeta 1}\Delta t/8}e^{iL_{G1}\Delta t/4}e^{iL_{v\zeta 1}\Delta t/8}\right]e^{iL_{G2}\Delta t/4} \end{aligned} \tag{4.33}$$

The algorithm which has to be programmed is now fully defined by equations (4.31) and (4.33). The parts of the operator decomposition are now applied on f

$$e^{iL_{NHC}\Delta t}f\left[\mathbf{r}^N, \mathbf{v}^N, \zeta_1, \nu_{\zeta_1}, \zeta_2, \nu_{\zeta_2}\right] \tag{4.34}$$

The Trotter expansion allows the sequential application of each of the terms. Before that, we can check the effect of the application of each part of the operator on our initial conditions. As an example we apply the first part of the expansion (4.33), iL_{G2}, on our initial conditions

$$\begin{aligned} &exp\left(\frac{\Delta t}{4}G_2\frac{\partial}{\partial\nu_{\zeta_2}}\right)f\left[\mathbf{r}^N, \mathbf{p}^N, \zeta_1, \nu_{\zeta_1}, \zeta_2, \nu_{\zeta_2}\right] \\ &= \sum_0^\infty \frac{(G_2\Delta t/4)^n}{n!}\frac{\partial^n}{\partial\nu_{\zeta 2}^n}f\left[\mathbf{r}^N, \mathbf{p}^N, \zeta_1, \nu_{\zeta_1}, \zeta_2, \nu_{\zeta_2}\right] \\ &= f\left[\mathbf{r}^N, \mathbf{p}^N, \zeta_1, \nu_{\zeta_1}, \zeta_2, \nu_{\zeta_2} + G_2\Delta t/4\right] \end{aligned} \tag{4.35}$$

The effect of the first operator is to shift ν_{ζ_2} by $G_2\Delta t/4$. Similar rules can be derived for all other parts of the operator. The complete transformation rules for the decomposition

in Equation 4.33 are given in the following listing:

$$
\begin{aligned}
e^{(iL_{G2}\Delta t/4)} &\longrightarrow \nu_{\zeta_2} = \nu_{\zeta_2} + G_2\Delta t/4 \\
e^{(iL_{\nu\zeta_1}\Delta t/8)} &\longrightarrow \nu_{\zeta_1} = e^{(-\nu_{\zeta_2}\Delta t/8)}\nu_{\zeta_1} \\
e^{(iL_{G1}\Delta t/4)} &\longrightarrow \nu_{\zeta_1} = \nu_{\zeta_1} + G_1\Delta t/4 \\
e^{(iL_{\zeta}\Delta t/2)} &\longrightarrow \zeta_1 = \zeta_1 - \nu_{\zeta_1}\Delta t/2 \\
e^{(iL_{C\nu}\Delta t/2)} &\longrightarrow \nu_i = e^{(-\nu_{\zeta_1}\Delta t/2)}\nu_i \\
e^{(iL_{\nu}\Delta t/2)} &\longrightarrow v_i = v_i + F_i\Delta t/(2m) \\
e^{(iL_{r})\Delta t} &\longrightarrow r_i = r_i + v_i\Delta t
\end{aligned}
\tag{4.36}
$$

Implementation of the Nóse-Hoover thermostat can now be done by applying the above equations successively according to the order defined by (4.31) and (4.33). The application of the whole chain of transformation rules can be separated into two parts. The variables $\mathbf{v}_i$ and $\mathbf{r}_i$ are changed during the application of the non Nóse-Hoover part of the Liouville operator, whereas the Nóse-Hoover part changes ζ_k, ν_{ζ_k} and $\mathbf{v}_i$. Due to this, the separation of the algorithm into a part which changes positions and velocities, and a part which applies the Nóse-Hoover chains is possible. The thermostat can be used with any implemented method in CAST. Input variables are the switch to turn it on or off as well as the desired temperature. In pseudo code the implementation looks like follows

Algorithm 6 MD simulation with constant temperature

```
1: procedure NVT SIMULATION
2:     for each integration step do
3:         apply transformation rules of (4.36) according to Equation (4.33)
4:         calculate half step velocities and new positions
5:         calculate new atomic forces
6:         calculate full step velocities
7:         apply transformation rules of (4.36) according to Equation (4.33)
8:     end for
9: end procedure
```

Pressure The usual isochoric and adiabatic simulations solving Newton's equations of motion at constant volume cannot be used if dissipative non-equilibrium systems are studied, for example to obtain transfer properties. Also in equilibrium simulations, especially if long range interactions are involved and a potential truncated at a cutoff

radius is used, unavoidable slow drifts may occur that need corrections. Furthermore, if someone wants to obtain the Gibbs free energy, a simulation under constant pressure and temperature is necessary. Therefore, the availability of methods for constant pressure and temperature dynamics is of great importance. To be able to perform such simulations algorithms for constant pressure are necessary. In order to obtain control over the pressure, a Berendsen barostat [54, 55] has been implemented.

Pressure is defined by Clausius virial theorem:

$$\Xi - 3PV + 2E_{kin} = 0 \Longrightarrow P = \frac{2}{3V}(E_{kin} - \Xi) \tag{4.37}$$

Here Ξ denotes the virial tensor, P the pressure, E_{kin} the kinetic energy and V the volume. Coupling to a constant pressure bath can be accomplished by adding an extra term to the equations of motion

$$\left(\frac{dP}{dt}\right)_{bath} = \frac{P_0 - P}{\tau_p} \tag{4.38}$$

where P_0 is the target pressure, P the current pressure and τ_p a coupling constant. A pressure change is realized by scaling of the inner virial tensor via scaling of the inter particle distances, with v as the particle velocity, α the scaling factor and x the particle position. The equations then read

$$\dot{x} = v + \alpha x \tag{4.39}$$

$$\dot{V} = 3\alpha V \tag{4.40}$$

The change in pressure is also related to the isothermal compressibility β, which is stated by the following relation

$$\frac{dP}{dt} = -\frac{1}{\beta V}\frac{dV}{dt} = -\frac{3\alpha}{\beta} \tag{4.41}$$

With this, α can now be determined by

$$\alpha = -\beta\left(P_0 - P\right)/3\tau_p \tag{4.42}$$

Combining Equations (4.42) and (4.38) and inserting into Equation (4.39) yields for

the modified equation of motion

$$\dot{x} = v - \frac{\beta(P_0 - P)}{3\tau_p} x \tag{4.43}$$

(4.43) represents a proportional scaling of the volume, the box length and the inter particle distances if one assumes an isotropic system. Therefore, at each time step, the scaling leads to $x \to \mu x$ and $l \to \mu l$. The scaling factor with first order in $\triangle t$ reads:

$$\mu = \quad 1 - \tfrac{\beta \triangle t}{3\tau_p}(P_0 - P) \tag{4.44}$$

$$\mu = \quad \left[1 - \tfrac{\triangle t}{\tau_p}(P_0 - P)\right]^{1/3} \tag{4.45}$$

It has to be noted, that the pressure is not a scalar (except for the isotropic case) but a second order tensor.

$$P = \begin{pmatrix} P_{xx} & P_{xy} & P_{xz} \\ P_{yx} & P_{yy} & P_{yz} \\ P_{zx} & P_{zy} & P_{zz} \end{pmatrix}$$

If the simulation box is a cube, the pressure can be calculated as the trace of the tensor:

$$P = Tr(\mathbf{P})/3 \tag{4.46}$$

In case of anisotropic box dimensions the equations have to be modified. In this case the volume V is the determinant of a matrix h which is formed by the column vectors **a**, **b**, **c** that represent the edges of the unit cell:

$$V = det\ h = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) \tag{4.47}$$

The scaling factor now becomes

$$\boldsymbol{\mu} = \mathbf{1} - \frac{\beta \triangle t}{3\tau_p} (P_0 - P) \tag{4.48}$$

The scaling of the particle coordinates $\mathbf{r}_i$ now becomes

$$\mathbf{r}_i^{'} = \boldsymbol{\mu} \mathbf{r}_i \tag{4.49}$$

and the scaling for the unit cell

$$h^{'} = \boldsymbol{\mu} \mathbf{h} \tag{4.50}$$

An example for the implementation together with the velocity Verlet algorithm is given in the following Algorithm 7

Algorithm 7 MD simulation with constant pressure

```
1: procedure CONSTANT PRESSURE SIMULATION
2:     for each integration step do
3:         calculate half step velocities and new positions
4:         calculate new atomic forces
5:         calculate full step velocities
6:         if constant pressure then
7:             calculate pressure from virial tensor
8:             calculate scaling factor
9:             scale box lengths and inter particle distances
10:        end if
11:    end for
12: end procedure
```

The pressure algorithm can be modified by several variables in the input file. Next to the on/off switch, the isothermal compressibility can be adjusted to match the used solvent. The standard value is set to 0.000046 bar^{-1} for water. Furthermore, the delay time in picoseconds for the barsotat can be set (standard = 2.0) as well as the target pressure in atmospheres.

4.3.4. Constraints

Constraints allow the use of bigger time steps, especially if they are used to freeze the fastest motions of the system. Several different constraint algorithms exist, depending on the integrations scheme used, but all are based on Lagrange multipliers. For the velocity Verlet integrator the RATTLE [166, 167] algorithm has been implemented. It is similar to the SHAKE [171, 172] algorithm which is one of the standard methods for performing such calculations. RATTLE calculates the positions and velocities at the next time step from the positions and velocities at the present step, without requiring information about the earlier history. It was developed specifically to work with the velocity Verlet algorithm and retains the simplicity of using cartesian coordinates for each of the atoms to describe the configuration of a molecule with internal constraints. The Lagrange multipliers are calculated in an iterative procedure which is displayed in Algorithm 8

Algorithm 8 RATTLE

```
1: procedure CONSTRAINT MD SIMULATION
2:     for each integration step do
3:         if RATTLE then
4:             while convergence not reached do
5:                 for all constraint bonds do
6:                     calculate r_ij^2
7:                     if r_ij^2 − rat^2 > 0.000001 then
8:                         calculate lagrange multiplier λ
9:                         update positions and velocities using lagrange multiplier
10:                    end if
11:                end for
12:            end while
13:        end if
14:    end for
15: end procedure
```

$$\begin{aligned} \vec{r_{ij}} &= \vec{r_i}(t + \triangle t) - \vec{r_j}(t + \triangle t) \\ r_{ij}^2 &= r_{ij}(x)^2 + r_{ij}(y)^2 + r_{ij}(z)^2 \\ \vec{d_{ij}} &= \vec{r_i}(t) - \vec{r_j}(t) \end{aligned} \tag{4.51}$$

In the first step the bond length of the constrained atom pair at time t (d_{ij}) and $t + \Delta t$ (r_{ij}) according to Equation (4.51) is calculated. The value of the quadratic bond length at time $t + \triangle t$ is compared to the desired bond length (rat) of the constraint $D = r_{ij}^2 - rat^2$. If the absolute difference (D) is smaller than a certain tolerance (0.000001 currently in the program) the next constraint is considered. If the difference is greater the algorithm searches for a Lagrange multiplier (λ) which satisfies the constraints more closely. It can be calculated by taking the dot-product of the bond lengths at t and $t + \triangle t$,

$$\Phi = d_{ij}(x)r_{ij}(x) + d_{ij}(y)r_{ij}(y) + d_{ij}(z)r_{ij}(z) \tag{4.52}$$

the reciprocal mass $1/m$ of atoms i and j and the difference between the desired and

the actual bond length. The Lagrange multiplier is given by

$$\lambda = \frac{D}{(2.0 \cdot (m_i + m_j) \cdot \Phi)} \tag{4.53}$$

With this Lagrange multiplier the velocities and positions are updated according to

$$\begin{cases} \textbf{Positions} : \vec{r}_i(t + \triangle t) = \vec{r}_i(t + \triangle t) - \lambda \cdot \vec{d}_{ij} \cdot \frac{1}{m_i} \\ \textbf{Velocities} : \vec{v}_i(t + \frac{1}{2}\triangle t) = \vec{v}_i(t + \frac{1}{2}\triangle t) - \lambda \cdot \vec{d}_{ij} \cdot \frac{1}{m_i \cdot \triangle t} \end{cases} \tag{4.54}$$

$$\begin{cases} \textbf{Positions} : \vec{r}_i(t + \triangle t) = \vec{r}_i(t + \triangle t) + \lambda \cdot \vec{d}_{ij} \cdot \frac{1}{m_i} \\ \textbf{Velocities} : \vec{v}_i(t + \frac{1}{2}\triangle t) = \vec{v}_i(t + \frac{1}{2}\triangle t) + \lambda \cdot \vec{d}_{ij} \cdot \frac{1}{m_i \cdot \triangle t} \end{cases} \tag{4.55}$$

Equations (4.54) and (4.55) show the update of the velocities and positions for atoms i and j after the calculation of the Lagrange multiplier. After the update, the loop is started again. This procedure is continued until all constraints are satisfied within the acceptable tolerance. The implementation allows to constrain specific bonds, not dependent on the atom type, and also a function to automatically constrain every hydrogen bond in the system. When a constant pressure simulation is to be performed, the constraints also contribute to the internal virial tensor. In this case the contributions are automatically calculated.

4.3.5. Boundary conditions

Spherical boundary conditions According to Hansen [173] a two dimensional system can be embedded in the surface of a sphere without the need for physical boundaries. This idea can be extended to three-dimensional systems which then form the surface of a hypersphere [174]. The curved geometry of the system may pose some problems due to the non-Euclidean geometry, but as the size of the system increases, these effects decrease and spherical boundary conditions become a valid method of simulating finite systems [175, 176]. When employing spherical boundaries, the system of interest is embedded in a sphere with a defined radius. Molecules or atoms who reach the sphere radius are perceiving a force which is directed at the center of the sphere to avoid vaporization during the simulation. The potential used for the boundary force is usually a harmonic potential of the form

$$E_{sphere} = k_s \left(|\vec{r}_i - \vec{r}_c| - r_s\right)^E \tag{4.56}$$

where k_s denotes the force constant in kcal/mol, $|\vec{r}_i - \vec{r}_c|$ the distance of the particle

from the geometric center of the sphere as displayed in Figure 4.7, $\vec{r}_s$ the radius of the sphere and E the exponent of the potential. The force applied on a particle is then

$$\vec{F} = E \cdot k_s \left(|\vec{r}_i - \vec{r}_c| - r_s \right)^{E-1} \cdot r_{i,c} \tag{4.57}$$

Thus a positive force constant will cause a force that moves atoms back in to the center of the sphere and a negative force constant will force atoms away from the sphere. The force is applied if the distance of a particle from the center exceeds the sphere radius. For simple harmonic boundary conditions an exponential factor of 2 or 4 is used and the value of the force constant is set to 10. The force is applied each time new gradients are calculated in the MD simulation. If desired, a second potential can be applied to simulate surface tension. The adjustable parameters in that case are the two radii and the two force constants of the potential.

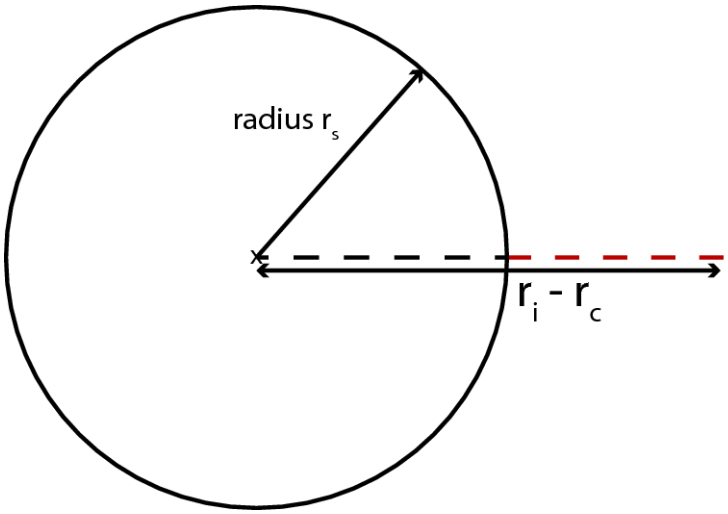


Figure 4.7.: Illustration of spherical boundary conditions. The sphere radius r_s is shown as the solid singke arrow. Distance of an arbitrary particle is shown as the dotted line. If the distance of the particle from the center exceeds the sphere radius (red part of the dotted line) the boundary potential is applied.

Periodic boundary conditions (PBC) [177–179] As the name implies, periodic boundary conditions can be used to simulate periodic systems. The basic idea is to choose the smallest possible unit cell in the system as the simulated system, and virtually copy it indefinitely in all directions to generate an infinite system. In the actual simulation only the conditions of the initial unit cell are calculated. The infinity is generated by the fact that particles that leave the simulation box reenter from the opposite side. PBC are usually used for the simulation of solvated macromolecules or

other mixtures, as well as bulk gases, liquids, crystals and solvated protein structures. Crucial parameters for the correct behavior of the boundary conditions are the size and shape of the unit cell as well as the chosen cutoff radius. The standard cell is a cubic cell with 90° angles at each corner and equal edges as depicted in Figure 4.8.

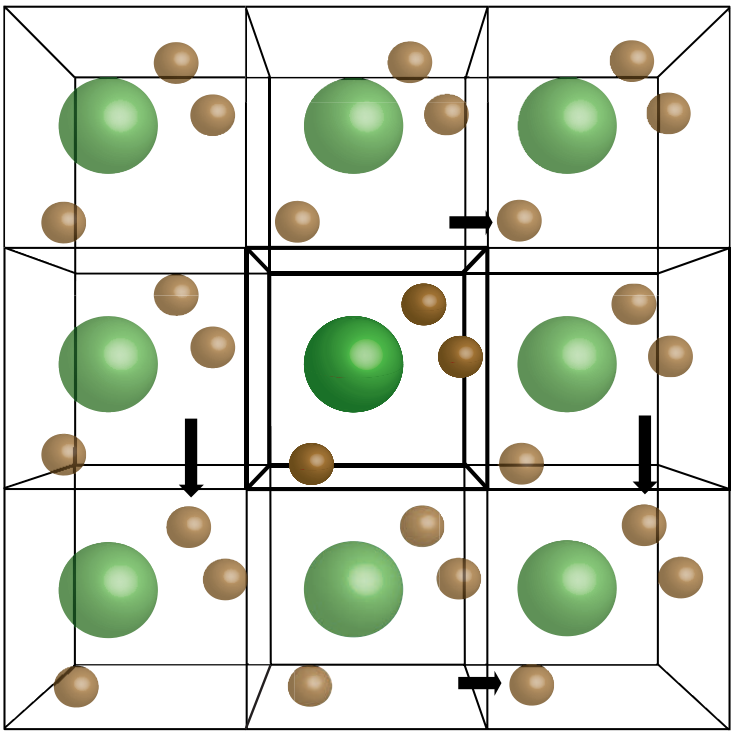


Figure 4.8.: Schematic representation of periodic boundary conditions. The central simulation cell is virtually copied in all dimensions. Particles only interact with the nearest neighbor of a potential interaction partner. Figure reproduced from *http : //isaacs.sourceforge.net/phys/pbc.html*

PBC can also be extended to support orthorhombic, triclinic and monoclinic shapes. CAST currently supports cubic and orthorhombic cell shapes.The choice of the cutoff radius is limited by the minimum image convention (MIC) [32]. The minimum image convention states that in a periodic system the cutoff distance may be no longer than half the shortest box vector. This ensures, that if chosen correctly, no atom can interact with a replicate of itself in another "fictitious" unit cell. Furthermore, no atom i can interact with two atoms j in different replicated cells. In short the MIC only considers interactions between a molecule (or particle) a and the closest periodic image of its neighbor. A simulation using PBC usually consists of the following steps seen in Algorithm 9.

Algorithm 9 Periodic boundary conditions

```
1: Set up unit cell centered on the origin
2: translate molecules into the unit cell
3: for each energy evaluation do
4:     for all particles p_a in cell C_a do
5:         for all particles p_b in cell b C_b do
6:             r^2 = ||x(p_a) − x(p_b)||^2
7:             if r^2 ≤ r_C^2 then
8:                 compute interaction between p_a and p_b
9:             end if
10:        end for
11:    end for
12: end for
```

At the beginning of each calculation the unit cell is built around the origin. Since it is not ensured that the system of question is also centered on the origin, all molecules are checked if they are within the designed simulation box. If this is not the case they are translated into the box. If a cutoff radius is used which is larger than half the box dimension the standard implementation cannot be used and the unit cell has to be replicated, which dramatically increases the simulation time. For the PBC the non-bonded energy function has been modified with a switch to check if PBC shall be computed. For every pair calculated in the non-bonded energy function the minimum image criterion is checked and only the interaction between i and the nearest of the 27 possible atoms j is calculated. Furthermore, after every complete cycle, the center of masses for all molecules are checked and, if necessary, the atoms are translated back into the simulation cell. Periodic boundary conditions are also compulsory for the use of the particle mesh Ewald method (see Section 4.2). In this case the standard interactions, modified according to the Ewald derivation, are calculated within the specified cutoff radius. All other electrostatic interactions within the unit cell are then calculated on the PME grid using fast Fourier transform algorithms.

4.4. Free energy algorithms

4.4.1. Umbrella sampling

As described in Section 3.4.1 umbrella sampling [71, 72, 142] is based on introducing a biasing potential. The CAST implementation features two possible reaction coor-

dinates, namely distances and torsion angles. The distance coordinate as well as the torsion coordinate is not limited to already existing bonds or torsions in the system, but can be applied to any two (or respectively 4) arbitrary particles in the system. Both reaction coordinates are restrained using a half harmonic bias potential, $V = \frac{1}{2}kx^2$. For the torsion angles the IUPAC [180] nomenclature is used, meaning that the torsion angle value reaches from 0°to 180° and 0° to −180°.

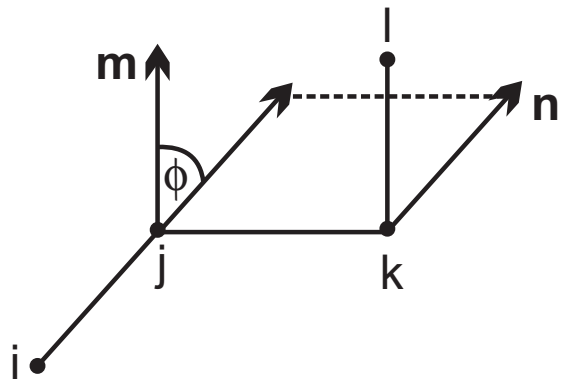


Figure 4.9.: Cross product definition of a torsion angle. The angle is defined by atoms i, j, k, and l. The sign can be calculated by the dot-product of vectors $\mathbf{r}_{ij}$ and $\mathbf{n}$

The IUPAC [180] convention for the sign of a torsion angle is depicted in Figure 4.9 and says that the sign of the angle is negative, if looking in the direction from k to j, the bond (k, l) must be rotated around the axis (k, j) counterclockwise to reach the cis-conformation over the smallest rotation angle. Mathematically the sign can be determined by the following expression [181]

$$sign(\Phi) \equiv signum\,(\mathbf{r}_{ij} \cdot \mathbf{n}) \tag{4.58}$$

This calculation is specifically done only in the biasing function. During normal force field execution it is neglected since a $0° - 360°$ definition is used. The bias potential is applied after every gradient evaluation during the simulation as long as the respective flag has been activated. It has been implemented as a standalone function and can be used with every underlying method supported by CAST including the MOPAC and TeraChem interfaces. The values for the distance restraint have to be given in Å, for torsion angles in degree (°). The force constants have the format $kcal/(mol \cdot Å^2)$ and $kcal/(mol \cdot degree^2)$.
Statistics of the simulation are controlled via the input file and are written to output files consistent with the format for post processing with WHAM. Benchmark calculations can be found in Section 5.1.3.

4.4.2. Free energy perturbation

The FEP formalism (for details see Section 3.4.2) has been implemented according to the original method proposed by Zwanzig [182]. Performing alchemical transformations requires the definition of the different topologies of the reference and target state. Two approaches can be found in the literature for this purpose in which the reference state, target state and all intermediate states are described by either a single or two separate topologies.

Single topology approach [183, 184] In the single topology approach, one uses a topology which can be found in both, the initial and the target state, of the simulation. Usually the more complex state serves as the blueprint for the topology. If particles are missing in the final state, ghost particles have to be introduced. During the simulation, the ghost particles have their non-bonded parameters successively scaled to zero and vanish. An example can be seen in Figure 4.10. The more complex topology is the one with the CH_3 group and is therefore used as the starting point for the common topology. During the mutation, one carbon atom is turned into a hydrogen and three hydrogen atoms have to vanish. Here, the three hydrogen atoms are turned into ghost particles and have their non-bonded interactions scaled down to zero as the order parameter λ, which controls the progress of the transformation moves from 0 to 1. Scaling of the force field parameters is usually done by

$$q_i\ (\lambda) = \lambda\ q_i + (1-\lambda)\ q_i \tag{4.59}$$

$$R_{ij}\ (\lambda) = \lambda\ R_{ij} + (1-\lambda)\ R_{ij} \tag{4.60}$$

$$\epsilon_{ij}\ (\lambda) = \lambda\ \epsilon_{ij} + (1-\lambda)\ \epsilon_{ij} \tag{4.61}$$

Here λ denotes the order parameter, q the partial charges of the particles and ϵ and R the force field van-der-Waals parameter. A critical point in the single topology approach is the modification of the chemical bonds during the transformation, since the bond length and the force constants change. The free energy changes which correspond to such bond changes are expected to cancel out in most simulations [185]. The same goes for the affected angles although these contributions can be calculated if needed, for example if the bonds and angles are strongly deformed by steric hindrances [186].

Dual topology approach [185–187] Contrary to the single topology paradigm, the dual topology approach features both states of the alchemical transformation simulta-

neously during the simulation. A comparison of both approaches is depicted in Figure 4.10.

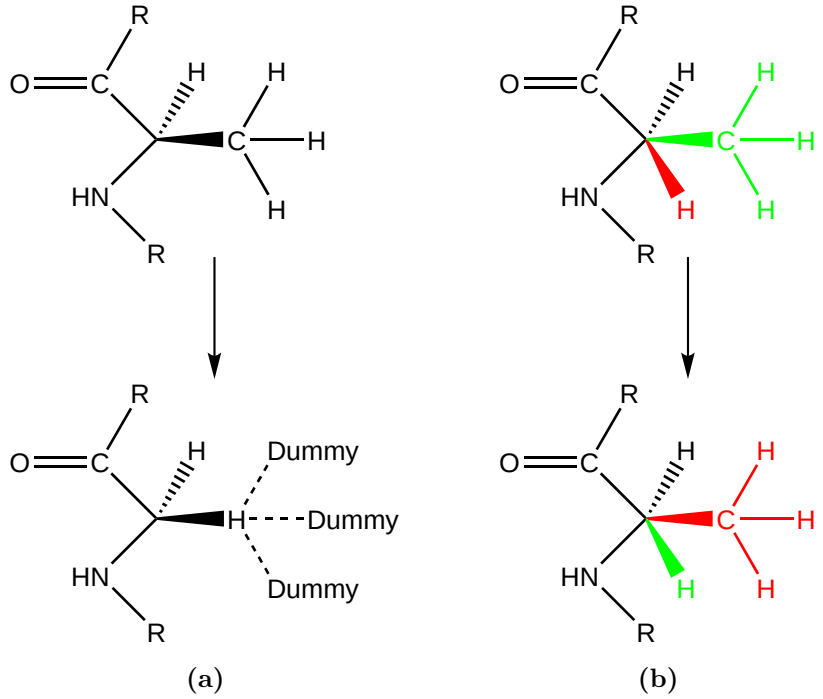


Figure 4.10.: Comparison between single topology (a) and dual topology (b) approach. In the single topology approach all atoms of the sidechain have to be eliminated by scaling their van-der-Waals and point charge parameters to zero. Furthermore, the carbon atom has to be transformed into a hydrogen by successively altering the van-der-Waals and charge parameters. In the dual topology approach both sidechains are always present, without seeing each other. The scaling of the topologies is realized by the order parameter λ. Green atoms mark the phased-in atoms, red the phased-out atoms.

A strict condition of this scheme is that the atoms of the final and initial state don't interact during the simulation. This can be achieved by using an exclusion list or just leaving out these atom pairs during build up of the non-bonded list. Furthermore, the interactions, intra- and intermolecular, of these moieties are scaled by the order parameter λ. At the starting point of the calculation ($\lambda = 0$) only the atoms of the initial state interact with the rest of system. At the end point only the atoms belonging to the target system interact with the system. The scaling of the potential energy U of the system can be described as follows

$$U(x;\lambda) = \lambda U_1(x) + (1-\lambda)\, U_0(x) \tag{4.62}$$

where $U_0(x)$ and $U_1(x)$ are the potential energies of the different states.

The dual topology approach has some advantages over the single topology approach. First, no chemical bonds have to be altered during the simulation. Second, the manipulation or scaling of non-bonded parameters, inherent to the force field, is also not needed. Yet there are some drawbacks. When the simulation reaches one of the end points ($\lambda = 0, 1$), it can become numerically unstable which is referred to as "end-point-catastrophes". These end points feature very weak yet non-zero interactions between the target or initial states and the environment. This can lead to cases where some of the vanishing or appearing atoms clash against already existing particles which in return can lead to large fluctuations in $\langle \Delta U \rangle$. These fluctuations can only be attended to by very extensive sampling. Another problem may be the flexibility of the transformed groups. If conformational changes are possible, not only one but several conformations may be contributing to the free energy change.

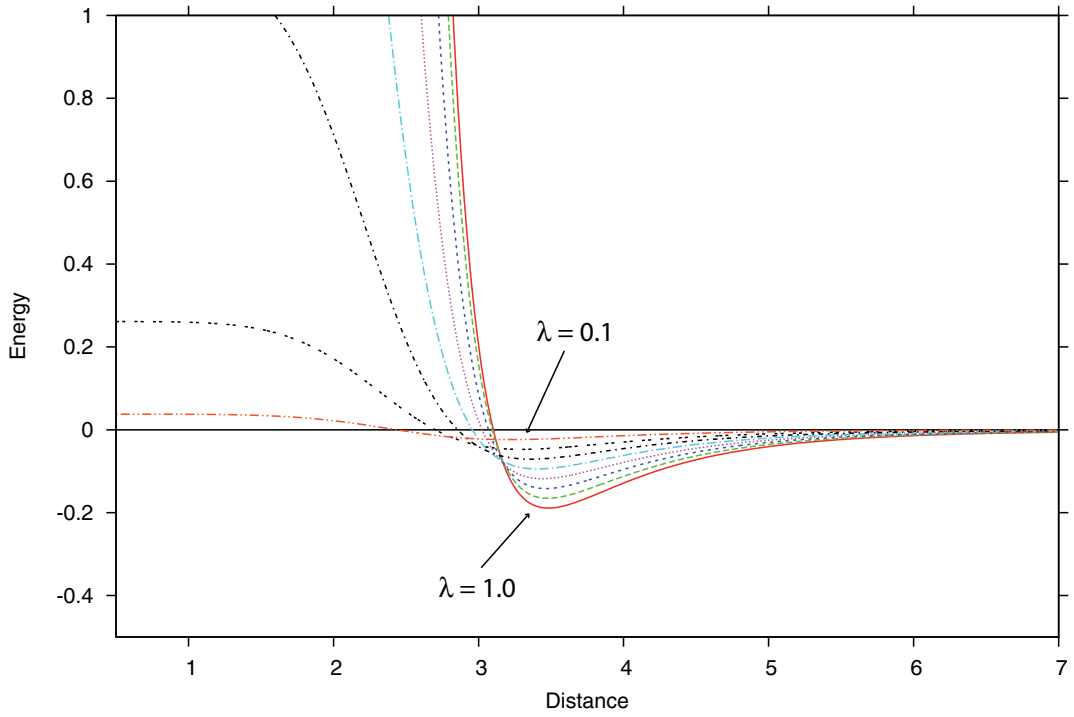


Figure 4.11.: Effect of the soft-core parameters on a Lennard-Jones potential for λ values between 0 and 1 in 0.1 steps. With increasing λ, the curve gets bounded from above for very small inter particle distances. Distance and energy are given in arbitrary units.

To avoid these problems, several strategies have been devised. Next to the use of non-linear mixing functions [188–190] or analytical fitting [191] the most widely used method is a modification of the parametrization of the van-der-Waals and charge term in the

potential energy function [100, 192, 193] that governs the interaction of an appearing or disappearing atom i with an unaltered one, j. Several schemes have been devised for this. Here one of the first approaches for the the van-der-Waals energy is given

$$U_{ij}^{LJ}(r^{ij}\lambda) = \lambda^n 4\varepsilon_{ij}\left(\frac{\sigma^{12}}{(\alpha_{LJ}\cdot(1-\lambda)\cdot\sigma^6+r^6)^2} - \frac{\sigma^6}{(\alpha_{LJ}\cdot(1-\lambda)\cdot\sigma^6+r^6)}\right) \tag{4.63}$$

Here, α_{LJ} is a positive constant, and σ_{ij} and ε_{ij} are the usual Lennard-Jones parameters found in force fields. The role played by the $\alpha_{vdW}(1-\lambda)$ in the denominator is to eliminate the singularity of the van-der-Waals interaction. The introduction of this soft-core potential results in bounded derivatives of the potential energy function when λ tends towards 0.

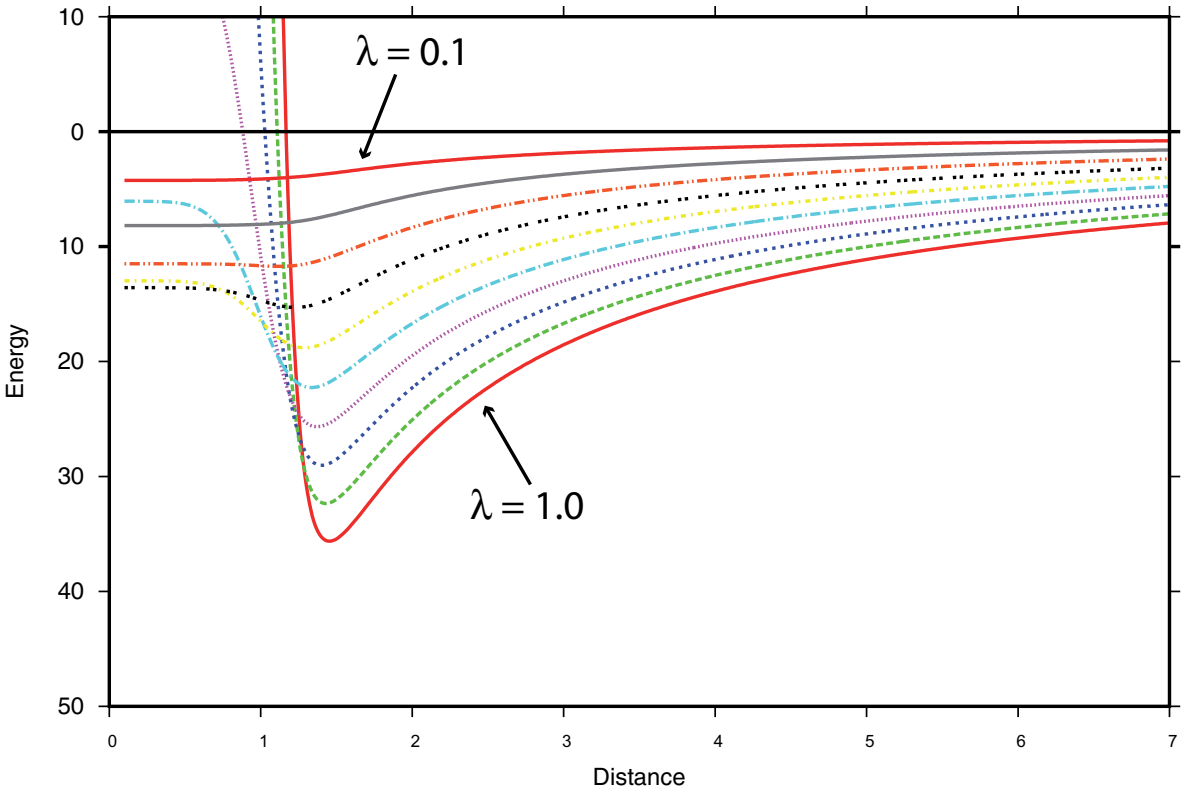


Figure 4.12.: Effect of the λ parameter on the combined electrostatic and van-der-Waals interactions. The curve gets smoothed for small distances to avoid big jumps in energy if two atoms come too close to each other. Distance and energy are given in arbitrary units.

In a similar manner the electrostatic potential can be treated as

$$U_{ij}^{C} = \lambda^{n} \cdot \frac{q_i q_j}{\sqrt[6]{r^6 + \alpha_{charge} \cdot (1-\lambda)}} \tag{4.64}$$

Here, α_{charge} is a positive constant, and q_i and q_j are the partial charges of atoms i and j found in force fields. Combining these two potentials results in a λ dependent potential illustrated in Figure 4.12. For the CAST implementation the van-der-Waals soft-core potential has been modified in accordance with the suggestion of Zacharias et al. [194] by

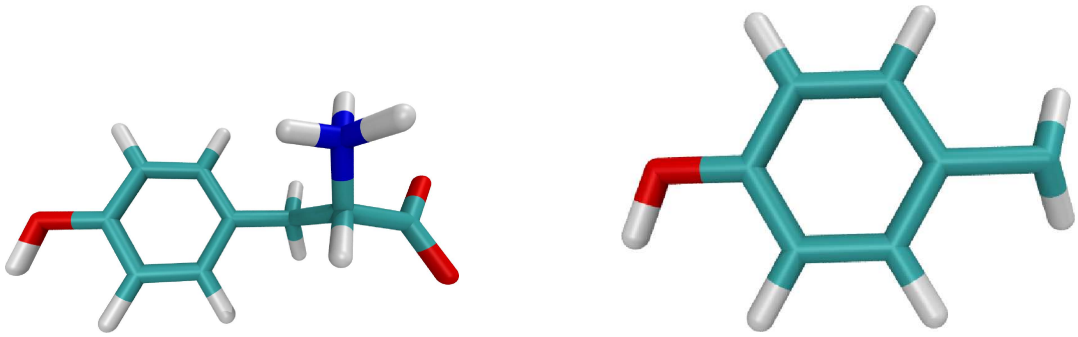
$$U_{ij}^{LJ}(r^{ij}\lambda) = \lambda 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}^2}{r_{ij}^2 + \alpha_{LJ}\cdot(1-\lambda)}\right)^6 - \left(\frac{\sigma_{ij}^2}{r^2 + \alpha_{LJ}\cdot(1-\lambda)}\right)^3\right] \tag{4.65}$$

The electrostatic potential is scaled linearly by λ

$$U_{ij}^{C} = \lambda^{n} \cdot \frac{q_i q_j}{r_{ij}^2} \tag{4.66}$$

For the definition of the topologies, CAST uses the dual topology approach. Based on the above definition, CAST has to be able to read chimeric systems with unusual number of bonding partners for atoms, for example a 5-bonding carbon atom. Furthermore, the chimeric system must be created and the exclusion list must be built during the program start up. To tell the program which atoms belong to the initial and final state, the atoms can be marked with an IN or OUT flag in the *xyz* coordinate file after the definition of the last bonding partner. Atoms marked with IN are considered atoms for the initial state, atoms marked with OUT belong to the target state. During build up of the interaction lists (bonds, angles etc.), interactions between atoms marked IN and OUT are automatically neglected. For the build up of the chimeric coordinate file, two supplementary programs have been written which are called CUT and ADD. CUT enables the user to delete single or multiple atoms from a given coordinate file and restores the "cut" structure to a readable file with new and corrected connectivities. For this, the program reads two files. The first file has to be named *delete* and contains a series of integers divided by space, the second file that has to be provided is the file containing the structure from which the substructure is cut from. The integers are the index numbers of the atoms which are to be deleted. The output of the program is a coordinate file named *final.xyz* which contains the new structure with the rearranged index numbers and topology. As an example let us look at a transformation from alanine to tyrosine in a trialanine. Starting from the tyrosine (Figure 4.13 (a)) the ter-

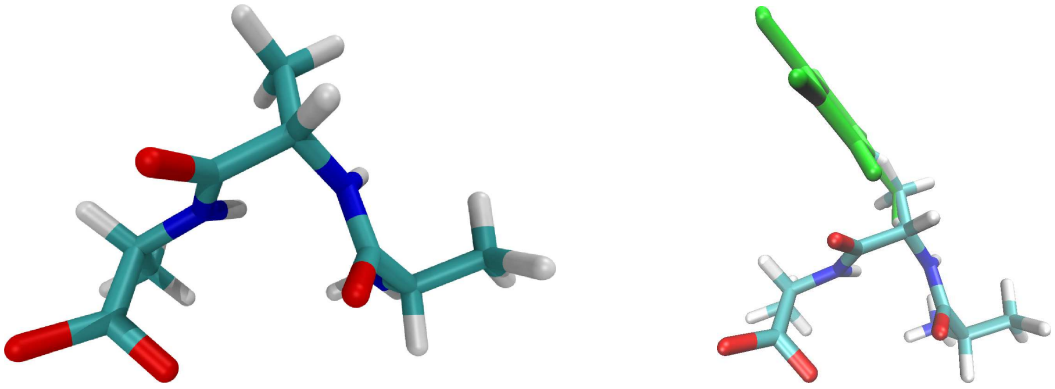
minal nitrogen and the respective backbone and hydrogen atoms are removed from the structure so that only the defining sidechain remains (Figure 4.13 (b)). This remaining sidechain is then attached to the alpha carbon of the middle alanine of the trialanine.



(a) **(b)**

Figure 4.13.: Illustration of the CUT procedure. Form the complete tyrosine (a) the first 6 atoms are cut, which eliminates the NH_3 as well as the carboxyl group. The resulting system is shown in (b). The atom forming the new bond in the chimeric system will be the carbon atom of the remaining CH_2 group.

ADD connects the remaining atoms from the cut structure with the main system to form the chimeric system for the FEP calculation in a single coordinate file. Taking the remaining system from Figure 4.13 (b) the atoms are attached to the middle α carbon of the trialanine. The resulting chimeric system possesses a formally 5 binding carbon at the central position as depicted in Figure 4.14 (b). ADD expects several inputs from the user which have to be entered via the command line when the program is executed. The first two commands are the reference geometries and the transform state. The reference geometry is the system where the cut structure is to be added, the transform state is the cut structure. Both input structures have to be in TINKER format. The next three input variables are of integer type and refer to the atom of the cut structure which forms the new bond, the atom of the reference structure where the cut structure is attached to, and the atom which defines the direction of the newly formed bond. Taking the structure from Figure 4.13 (b) the first integer would be the carbon of the remaining CH_2 group of the cut tyrosine. The second integer would be the α carbon atom of the middle alanine from Figure 4.14 (a), and the third integer would be the index number of the carbon atom of the CH_3 side chain of the middle alanine.



(a) (b)

Figure 4.14.: Chimeric structures for the transformation of the tripeptide ALA-TYR-ALA to ALA-ALA-ALA. The natural tri-alanine peptide is depicted in (a) whereas (b) shows the chimeric structure constructed with the ADD and CUT programs. The middle alanine residue carries a formal 5-bonding α-carbon atom where in addition to the normal CH_3 side chain, the tyrosine side chain is attached.

FEP runs The CAST implementation of the FEP formalism features a fully automated process for an alchemical transformation. Provided the correct coordinate file is present, the complete PMF over all windows can be calculated with only one program start. An overview of the general steps during a CAST FEP calculation can be seen in Figure 4.15. The total free energy change for each window is written after every window has been finished. The detailed information about potential energy differences between the two states, divided into van-der-Waals and electrostatic, as well as the current temperature are written after every equilibration and production run. Runs are started by default at $\lambda = 0$. Only the final value and the value for the increments have to be supplied. The subroutine automatically calculates the necessary number of windows and loops through the whole transformation. Phasing in and out of the atoms is controlled by the λ parameter. For further control of the transformation the scaling of van-der-Waals and electrostatic potentials can be controlled individually. It has been shown that, in case of appearing or disappearing atoms, simultaneous modification of the electrostatic and van-der-Waals terms in the potential energy function leads to numerical instabilities in the MD trajectories, especially if the system contains vanishing atoms [93,94]. When the van-der-Waals parameters of these atoms become quite small, that is when λ is approaching 0 or 1, they can come extremely close to other particles in the system. Since the vanishing atoms still carry residual charges, the resulting non-bonded interactions often increase dramatically, which is incompatible with a perturbative approach. Therefore, the scaling of van-der-Waals and charge interactions are

decoupled from one another. The value for the scaling of the interactions can be set in the input file. Further user definable variables for the FEP control are shown in Table 4.2.

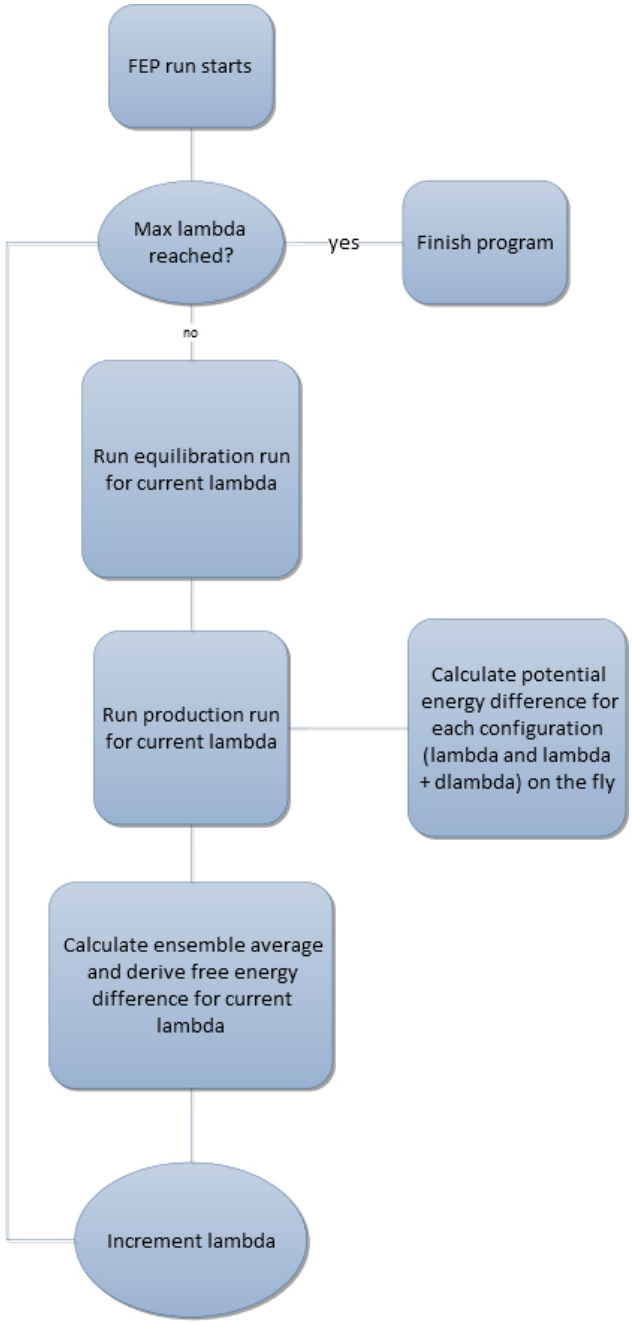


Figure 4.15.: Flowchart of a standard FEP calculation. After initialization the transformation is started at the λ value provided by the user (normally 0). For the current window, equilibration and production runs are performed and data is gathered. After production for the current window has finished λ is increased by $d\lambda$ which is also provided by the user. If λ has reached 1, the runs i complete and the program is finished.

Input parameters	Type	Description
FEPlambda	Float	Final value for the order paramater (max = 1)
FEPdlambda	Float	Increment of the order parameter
FEPvdwcouple	Float	start of van der Waals coupling
FEPeleccouple	Float	start of electrostatic coupling
FEPvshift	Float	value for the vdW softcore parameter
FEPcshift	Float	value for the charge softcore parameter
FEPequil	Int	number of equilibration steps for each window
FEPsteps	Int	number of production steps for each window
FEPfreq	Int	output frequency (each FEPfreq steps) for data

Table 4.2.: FEP variables which can be controlled by the user.

4.5. PDB converter (CHARMM parameters)

Many programs use their own type of coordinate files regardless of some standardized formats like *xyz*, *pdb* or *mol2* files. CAST makes use of a special type of *xyz* files, first provided by the TINKER [195] program. The advantage of this format is the full definition of the topology compared to the standard *xyz* format. In the standard format only the cartesian coordinates for the atoms are provided. The TINKER format extends this to the respective bonding partners of the atoms. Here the topology is already fully defined, and the decision if an atom is bonded to another does not have to be decided on the calculation of the distance and comparison with standard bond length. Information on proteins on the other hand are usually provided in the special format of the Protein Data Base (PDB). These *pdb* files normally contain the index number of the atom, the type of atom, the amino acid it belongs to as well as the cartesian coordinates. If a calculation with CAST is to be performed the *pdb* file has to be converted to a *xyz* file in TINKER format. For this reason a conversion program has been written which automates this process, especially with FEP calculations of chimeric systems and ligand-protein calculations in mind. The current version makes use of the freely available program Visual Molecular Dynamics (VMD) [196] and is designed to generate TINKER coordinate and parameter files for the CHARMM force field. Conversion for AMBER parameters has been implemented by Peter Friesen during his Bachelor thesis [197].
The PDB-converter has been designed as a standalone program but can easily be integrated into CAST. The current version features 3 distinct conversion possibilities:

1. Convert a single protein
2. Convert a single *pdb* file generated with the SwissParam server
3. Combine two different *pdb* topologies

1. Converting a protein. As mentioned before the *pdb* files of the protein data base usually do not contain topology information (bonds, angles, torsions etc.). Therefore, this information has to be generated by other means. The standard procedure for a calculation with the CHARMM force field is the Automatic-PSF-Builder implemented in VMD. The *psf* file contains all topology information including bonds, angles, cross-terms, torsion angles, and improper dihedrals. Using the *psf* builder requires the following files, the *pdb* file from the data base, the original CHARMM parameter file containing the force field parameters and the original CHARMM *rtf* file containing atom type definitions. In a first step, these files are now used to generate the *psf* file. After that the CHARMM converter comes into play. With the *pdb*, *psf*, *rtf* and parameter file the converter is able to generate a complete TINKER style coordinate file and a TINKER style force field parameter file. A flowchart can be seen in Figure 4.16. In TINKER-like parameter files, atoms are assigned a type as well as a group. Covalent interactions as well as van-der-Waals interactions are stored between atom groups, charges are stored for each atom type. This reduces the total number of parameters and information which has to be stored in the file. The conversion can be done with systems of any desired size. Force field parameter generation has been implemented in a way that the program recognizes if an atom type has been previously defined. In that case the new atom gets assigned the same type and group.

2. Converting SwissParam output

Force fields are generally parametrized for a given set of atoms or atom types. Newly designed drugs or other compounds may impose the problem that some atoms do not have a parameter in the force field. In that case a possibility to generate such a parameter is needed. The AMBER program, next to the standard AMBER force field, provides a generalized amber force field (GAFF) [198] and the ANTECHAMBER [199] program with which almost any type of atom can be parametrized. These new parameters are in line with the standard AMBER parameters. For the CHARMM force field such generation of parameters can be done with the SwissParam force field generation tool [200]. The server uses a *mol2* type input file to generate the *psf*, *pdb*, and *rtf* files needed for a calculation with CHARMM. The process for the conversion of a structure generated with SwissParam is essentially the same as for the standard *pdb* files. The

only differences are some minor formatting differences compared to the original *rtf* and parameter files.

3. Combining two topologies

Combining topologies is useful if the protein and the inhibitor can not be parametrized using the same parameters. In that case the inhibitor has to be cut from the protein structure and stored in a separate *pdb* file. The protein and the inhibitor can now be parametrized and converted separately (see 1. and 2.). After conversion of both parts to their own respective coordinate and parameter files, topology and force field parameter files can be merged to reproduce the original inhibitor/protein system, this time although in TINKER format. In this case the inhibitor is put at the end of the protein coordinate file. It has to be noted that no covalent bond can be generated between inhibitor and protein since that would again change the parametrization of the part involving the new covalent bond. Nevertheless, bond cleavage and formation can be done by manually editing the final coordinate file. The missing force field parameters can now also be added manually to the force field parameter file.

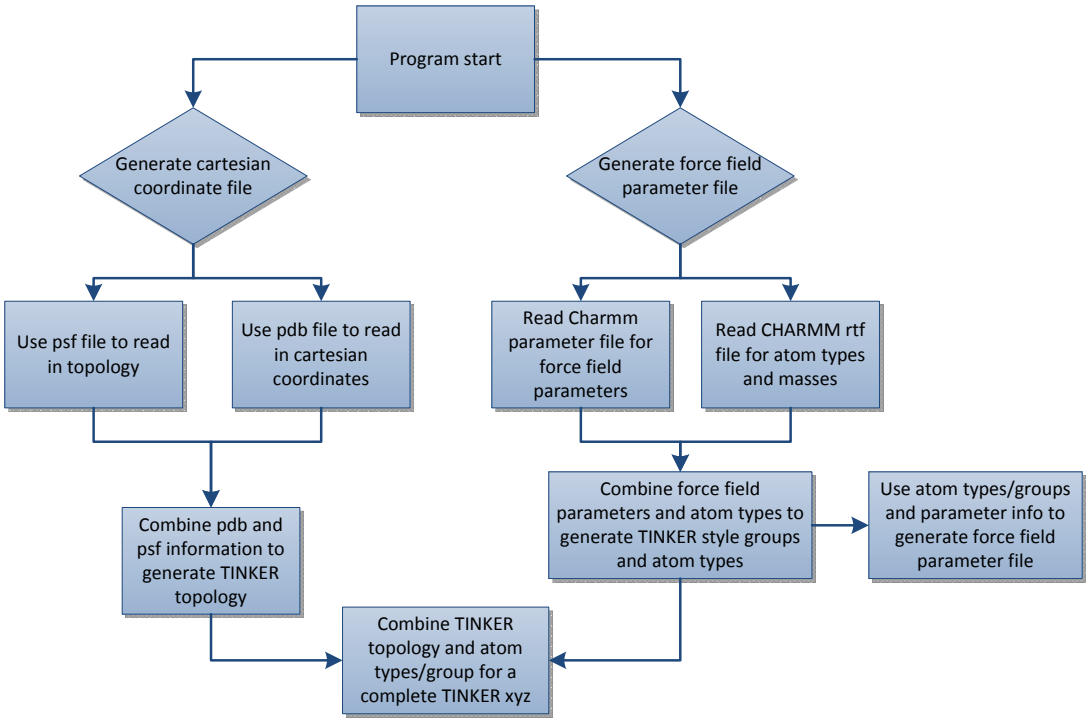


Figure 4.16.: Flowchart for the CHARMM converter. The original CHARMM files in combination with a *psf* file obtained with VMD are used to generate TINKER coordinate and force field parameter files.

4.6. MPI interface to TeraChem

Motivation A common strategy when performing global optimization is to first use a force field to generate the minimum structures and then optimize them using *ab initio* calculations. With this strategy, one can often achieve reasonable results, but depending on the force field quality description of the energetics may be poor. This in return can lead to the problem that important configurations or conformations may be missed during the sampling. Furthermore the number of structures that need refining gets large [201]. Another problem of force fields may be the fact that simply no parameters are available for the system of interest. In that case new parameters have to be generated which actually may take longer than the global search itself [150, 151]. Another drawback of force fields is the fact that they are not suitable for problems where bond breaking or bond making is involved which is often the case if for example protonation sates or reaction pathways are investigated. A way to circumvent this problem and to get more accurate results is to use *ab initio* methods like Hartree-Fock [122] or DFT [124] in combination with a sufficiently large basis set instead of force fields. The major problem here is that standard programs based on CPU architectures are far too slow to warrant a full global optimization. One possibility to increase the speed of such programs is to accelerate them with the massive stream processing capabilities of graphic processing units (GPU). A famous example is the program TeraChem [168,169] which is an *ab initio* electronic structure program that makes use of GPUs. Compared to standard programs, the use of GPUs provides a great speed up which allows the simulation of whole proteins using DFT methods which in return makes the program ideally suited for global optimization approaches.
The global optimization routines in CAST based on the TabuSearch can be divided into three basic steps. Those are the local optimization using a standard optimizer, the modest ascent part (neighborhood search or Dimer-method [202–205]) to leave the current minimum and the basin hopping algorithm. Since all these parts are hard coded in the CAST program, only energy and gradients are needed to progress the search procedure. Until now energy and gradients can be obtained by the internal force field implementation or via system call externally for semi-empirical or *ab initio* based calculations. In the case of system calls, the information needed by the external program as well as the results of the calculation are written to and read from the hard disk drive (HDD). Yet reading from and writing to the HDD is extremely slow and inefficient. A better way would be the combination of search and energy and gradient evaluation in one program but at the cost of a far more complex source code and extensive code maintenance. An alternative is presented by the Message parsing interface (MPI) [206] and its

"named port" facility. MPI allows the exchange of data between different threads. The most common application for MPI is the massive parallelization of programs. In that case the different threads are spawned by a single program and the work is distributed among the threads. The threads can now communicate through the interface and share their distributed information, for example the non-bonded interactions during a parallel force field calculation. In the presented work the MPI is used as a tool to standardize interprocess communication, that is a client-server model. The “named port“facility in MPI version 2 behaves like the socket interface in UNIX/C, but provides a standard interface which can be seamlessly accessed from any programming language with MPI support.

Implementation In the implemented approach, MPI manages the communication between two different programs. CAST acts as the client while TeraChem is the server. A detailed flowchart of the communication procedure is given in Figure 4.17. Both programs are started with MPI. The server (TeraChem) opens a communication port and publishes it to MPI using a pre-shared name. The client (CAST) is looking for the published port using that name and initiates the connection. After the connection is established successfully, CAST sends the general input information to TeraChem. When CAST requires an energy and/or gradient calculation it sends a message to TeraChem with molecular coordinates and waits for response. TeraChem performs the desired task and sends back the requested information. As mentioned above, CAST in principle only requires energy and gradient values for each point. The remaining tasks (modest ascent and local optimizations) could be done within CAST. Nevertheless, it is more efficient to use the local optimization subroutine of TeraChem, as orbital information can be stored in memory leading to a much faster convergence of the SCF iterations. This ultimately leads to a faster local optimization and consequently a faster global optimization. Therefore, the modest ascent is performed using CAST, but local optimizations are done in TeraChem. After the geometry has converged, TeraChem sends back the geometry of the optimized structure and the corresponding energy. The dimer method can be used for an accurate description of a reaction path. However, our strategy is to leave a given local minimum in as few steps as possible. This is achieved by large increments. Consequently, in our approach the transition state is not accurately determined nor is the modest ascent accurately followed. However, as shown recently, this strategy considerably accelerates the global optimization without sacrificing convergence to the global minimum [207].

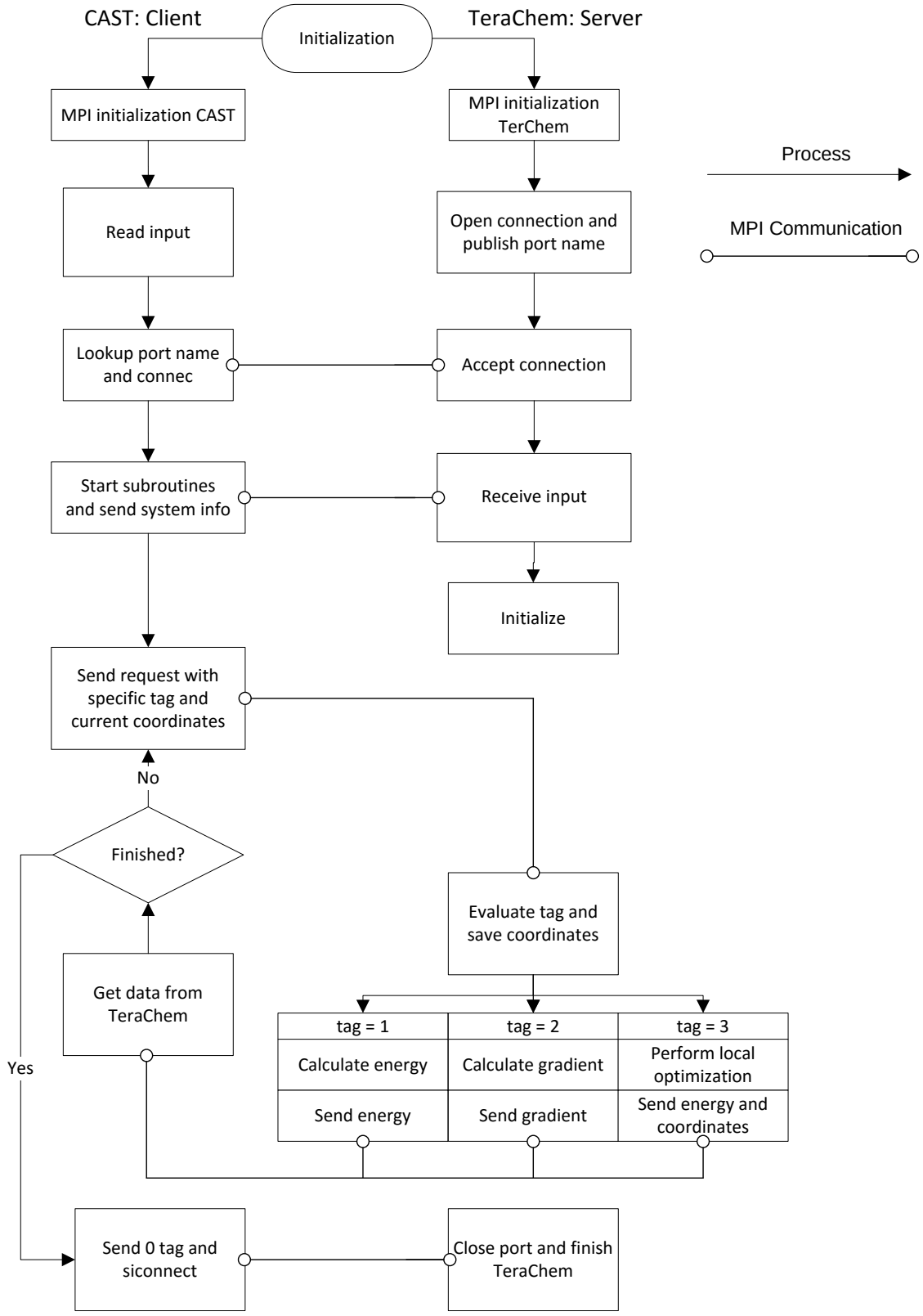


Figure 4.17.: Flowchart for the MPI communication between CAST and TeraChem. Both programs are initialized with MPI, CAST acts as the client, TeraChem is the server which accepts the connection from CAST.

Due to the large step size, occasionally the geometry optimization to the next local minimum starts from a strongly distorted geometry. To reduce computational time when starting from these distorted structures, our algorithm uses force field approaches for the first few steps of the optimization and then switches back to DFT for structure refinement. A first application and proof of concept can be found in Section 5.2.

4.7. Implicit solvation: Generalized Born (GB)

The following methods were implemented in the CAST program by Peter Friesen during his bachelor thesis. In the following chapter a short overview of implicit solvation methods, the Generalized Born (GB) method as well as the implementation by Peter Friesen is described. For further details please refer to his bachelor thesis [197].

Implicit solvation A lot of properties of substances depend on the nature of its environment, especially if the substance is dissolved or in vacuum. If the substance is dissolved the nature of the solvent also has a great influence. Since a lot of chemical reactions take place in solution, the accurate description of the solvent influence is essential. The description of the solvent can be divided into two groups: explicit [208,209] and implicit [210] solvent models. In the explicit model the actual solvent molecules and their interactions with the solute and other solvent molecules are described. Although this is the most straightforward and realistic method the computational power needed is very high due to the large number of solvent molecules that need to be simulated. An alternative is to substitute the actual solvent molecules with a continuum that simulates the dielectric properties of the solvent. This has some advantages:

- Far lower computational power is needed.
- There isn't need for equilibration since the continuum instantly reacts to changes.
- No description of viscosity speeds up conformational changes.
- Many local minima which arise due to geometrical changes in the solvation shell are neglected.
- Degrees of freedom of the solvent molecules are taken into account implicitly. This leads to a far more efficient estimation of the enthalpy of solution.

Since this is a strongly simplified description some drawbacks are imminent. Due to the neglect of the viscosity reaction kinetics cannot be described correctly. Furthermore, no

hydrogen bridges which are essential for a lot of properties can be described. Another problem are hydrophobic effects which arise due to entropic effects of the solute. Nevertheless quite a number of methods have been developed based on the implicit solvent scheme. Based on the cavity in the solvent generated by the solute, the enthalpy of solvation can be calculated with the electrostatic and van-der-Waals interactions between solute and solvent. This came to be known as the polarized continuum model (PCM) [211, 212]. Another possibility is to use induced charges of the solvent on the surface of the solute. This conductor like screening model (COSMO) [213] is nowadays a widely applied method. One of the most used schemes, due to its efficiency, is the generalized born (GB) [214,215] method. In the following a short description of implicit solvation and the generalized born model is given.
The basic assumption for most implicit solvent methods is the separability of the Hamiltonian given by

$$E_{tot} = E_{vac} + \triangle G_{solv} \tag{4.67}$$

with E_{tot} as the total energy, E_{vac} the potential energy in vacuum and ΔG_{solv} the free enthalpy of solvation. This mixing of potential and free energy is normally not permitted, but allows a description of the influence of the solvent on the solute. Another assumption is the splitting of ΔG_{solv} given by

$$\triangle G_{solv} = \triangle G_{cav} + \triangle G_{vdW} + \triangle G_{pol} \tag{4.68}$$

Here ΔG_{cav} is the energy needed to generate a cavity in the solvent, $\triangle G_{vdW}$ is the vdW interaction of the solute with the solvent molecules and $\triangle G_{pol}$ is a polarization contribution which describes the electrostatic interactions of solute and solvent in a dielectric medium [216]. ΔG_{cav} and $\triangle G_{vdW}$ can be combined to a non-polar contribution $\triangle G_{npol}$ resulting in

$$\triangle G_{solv} = \triangle G_{npol} + \triangle G_{pol} \tag{4.69}$$

The calculation of the non-polar contribution is normally based on the solvent accessible surface area (SASA) of the solute [217]. Then $\triangle G_{npol}$ yields

$$\triangle G_{npol} = \sum_i \sigma_i SA_i \tag{4.70}$$

Here SA_i is the solvent accessible surface area and σ_i is a solvation parameter which is obtained empirically by comparing results of calculations with explicit solvation. The computationally demanding part ΔG_{pol} is based on the effective born radius α [218,219]. The dissolving of a solute in a solvent in the generalized Born model can be described

by a thermodynamic cycle depicted in Figure 4.18.

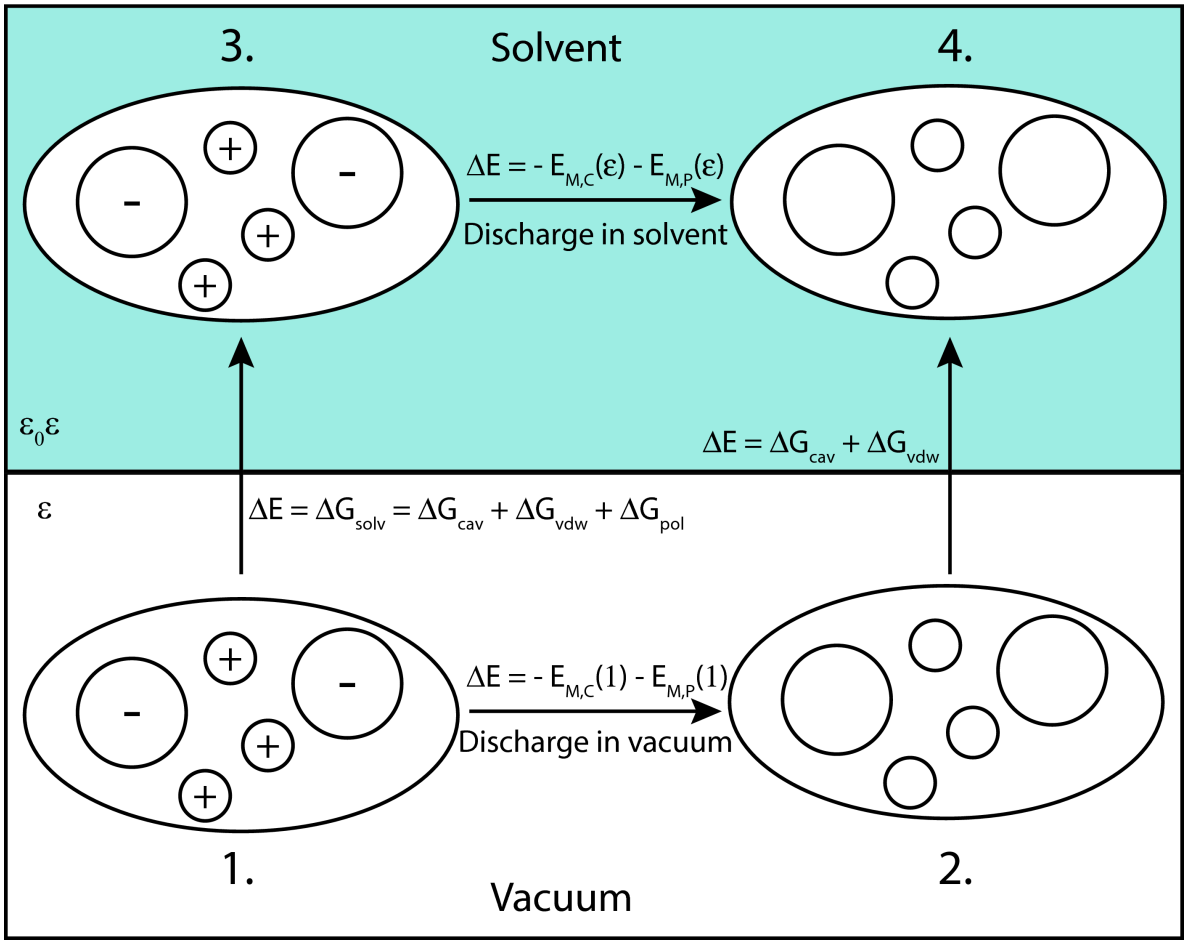


Figure 4.18.: Solvation of a molecule in a solvent depicted as a thermodynamic cycle. According to the GB model, the solvent has a relative permittivity ε. Atoms are depicted as circles with their partial charges. The solvent accessible surface area is depicted as the larger border around the molecules. Figure has been reproduced from ref [197] with permission from the author.

The contributions to the energy are $E_{M,C}$, the energy needed to charge a molecule in a dielectric medium and $E_{M,P}$, the energy needed to polarize the surrounding during the charging of the molecule. Thereby, $E_{M,C}$ and $E_{M,P}$ are defined as

$$E_{M,C}(\epsilon) = \frac{1}{8\pi\epsilon_0\epsilon} \sum_{i \neq j} \frac{q_i q_j}{r_{ij}} \tag{4.71}$$

$$E_{M,P}(\epsilon) = W_{\alpha_i,\epsilon}(q_i) = \frac{1}{8\pi\epsilon_0\epsilon} \sum_i \frac{q_i^2}{\alpha_i} \tag{4.72}$$

Here q are the partial charges of the particles, α the effective born radii, and ε and ε_0 the dielectric constants. Taking the cycle from Figure 4.18 the ΔG_{pol} part of the solvation energy amounts to

$$\begin{aligned}\Delta G_{pol} &= -\left(E_{M,C}(1) - E_{M,C}(\epsilon)\right) - \left(E_{M,P}(1) - E_{M,P}(\epsilon)\right) \\ &= -\frac{1}{8\pi\epsilon_0}\left(1-\frac{1}{\epsilon}\right)\sum_{i\neq j}\frac{q_i q_j}{r_{ij}} - \frac{1}{8\pi\epsilon_0}\left(1-\frac{1}{\epsilon}\right)\sum_i \frac{q_i^2}{\alpha_i} \\ &= -\frac{1}{8\pi\epsilon_0}\left(1-\frac{1}{\epsilon}\right)\sum_{ij}\frac{q_i q_j}{\sqrt{r_{ij}^2 + \alpha_i\alpha_j\delta_{ij}}}\end{aligned} \tag{4.73}$$

with δ_{ij} as the Kronecker delta. This equation is a more general form of the original Born equation. This general GB equation is normally modified to reproduce experimental data. One modification proposed by Still [216] is

$$\Delta G_{pol} = -\frac{1}{8\pi\epsilon_0}\left(1-\frac{1}{\epsilon}\right)\sum_{ij}\frac{q_i q_j}{f_{ij}^{GB}} \tag{4.74}$$

$$with \quad f_{ij}^{GB} = \sqrt{r_{ij}^2 + \alpha_i\alpha_j e^{-D}}, \quad D = \frac{r_{ij}^2}{4\alpha_i\alpha_j}$$

Equation (4.74) is an approximation to the Poisson equation for the free enthalpy of solvation of a molecule in a ion-free dielectric medium. The challenge when using the GB model is to find a formalism which allows the effective and correct estimation of the born radii. "Perfect" born radii can be obtained using the Poisson equation [220]. Since the solution of this equation is computationally very demanding, normally approximations are used to generate the born radii as close as possible to the "perfect" radii. A common method is the combination of the calculation of the solvent accessible surface area and the born radii. These are known as generalized born surface area (GBSA) methods. The calculation of the SASA can be done analytically using Gauss-Bonnets theorem [221], numerically using the approximation of the surface integral with arbitrary accuracy [222] and via approximate analytical methods [223]. The SASA can then be used to calculate the non-polar part of the enthalpy of solvent using equation (4.70). Furthermore, the SASA can be used to calculate the born radii. Several methods have been developed differing in accuracy and speed. The born radii are then used to obtain the polar part of Equation (4.69) using Equation (4.74).

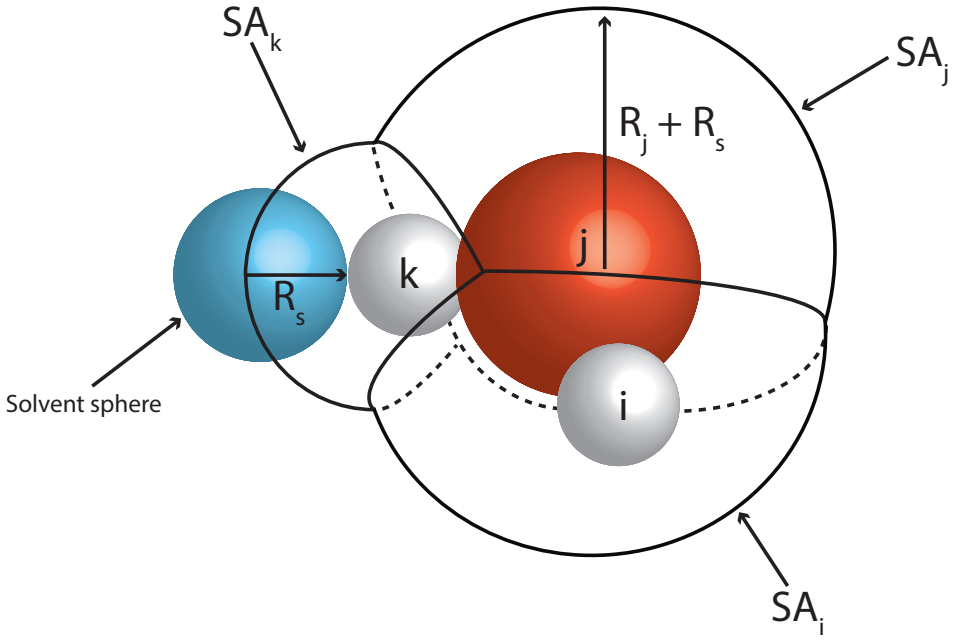


Figure 4.19.: Depiction of the solvent accessible surface area. Shown are the van-der-Waals surfaces of atoms i,j,k (red and white), their solvent accessible surface areas (SA_i, SA_j, SA_k)(edge) and a solvent sphere (blue). Figure has been reproduced from ref [197] with permission from the author.

Implementation The CAST software package has been extended by several methods for the calculation of implicit solvation energies based on the Generalized Born method. The source code has been extended by the classes *born* and *surface*.

Calculation	Implemented Method
SASA	Richmond, Gauss-Bonnet theorem [221]
Born Radii	numerical Still method [216]
	analytical Still method [224]
	HCT method [225]
	OBC method [226]

Table 4.3.: Overview of the newly implemented methods for the calculation of the enthalpy of solvation.

The class *born* includes the subroutines for the calculation of the born radii as well as the enthalpies of solvation and the respective derivatives in cartesian coordinates. The class *surface* contains the subroutines for the calculation of the solvent accessible surface area after Gauss-Bonnet. All implemented methods can be seen in Table 4.3. For the

verification of the implemented algorithms several test calculations were performed. The investigated systems were a pentapeptide and Ubiquitin (1UBQ). Results were compared with the TINKER program. For results see the bachelor thesis of Peter Friesen [227].

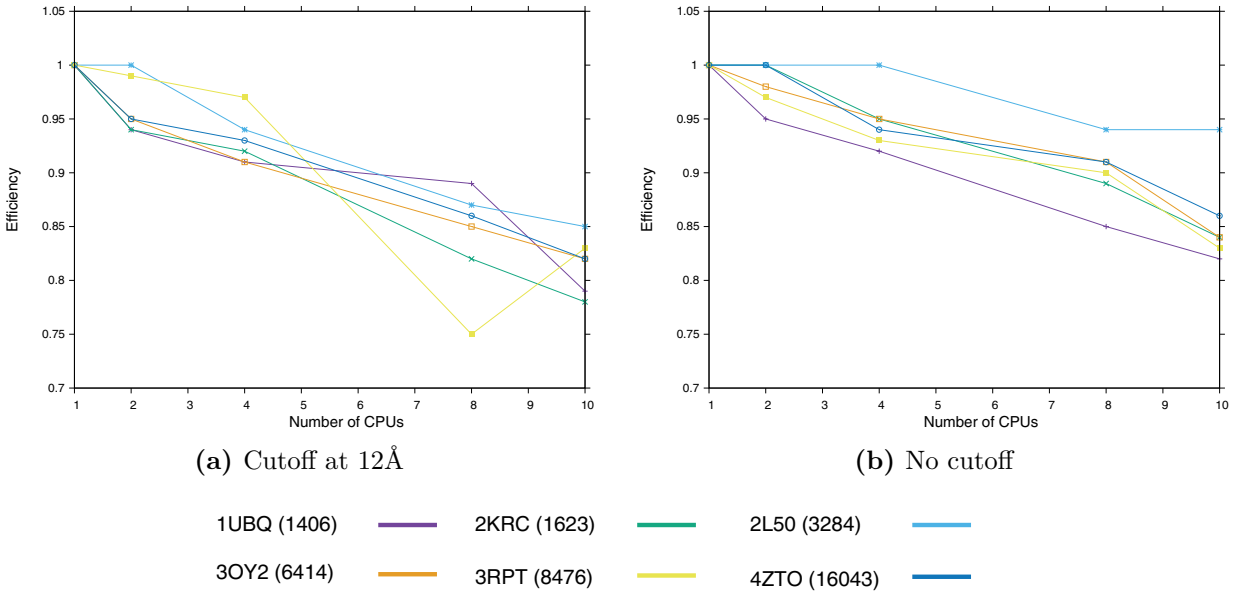
5. Application

5.1. Preliminary calculations

In Section 5.1 test calculations for some of the implemented algorithms as well as performance checks for the parallel implementation to verify the correct implementation and efficiency are presented. The parallel performance of the program is checked by single point and gradient calculations on several proteins of varying sizes. Furthermore, molecular dynamics simulations have been performed on the same systems. In Section 5.1.2 the linked cell algorithm is compared to the standard Verlet list build up routine for different cutoff radii. In Section 5.1.3 the umbrella sampling and FEP implementation are tested. For the umbrella sampling implementation distance and torsion restraints are tested by running two examples from literature and comparing the results. For the FEP implementation, three systems of the NAMD FEP tutorial are simulated with CAST and cross checked against the NAMD results.

5.1.1. Parallel performance

In the CAST program the OPLS-AA, CHARMM and AMBER force fields have been partly parallelized using the OpenMP programming model. The use of preprocessor directives allows the user to compile the same code with and without OpenMP support. Parallelization was done for the non-bonded interactions (van-der-Waals and electrostatics). The efficiency of the parallelization was tested on a 12-core Nehalem system (2.9 GHz) with 48GB RAM. Testsystems were various proteins, taken from the protein data base. To minimize the error due to memory initialization or I/O 100 consecutive gradient calculations were performed for each testsystem. Afterwards the average value for a single point, including gradients was taken.



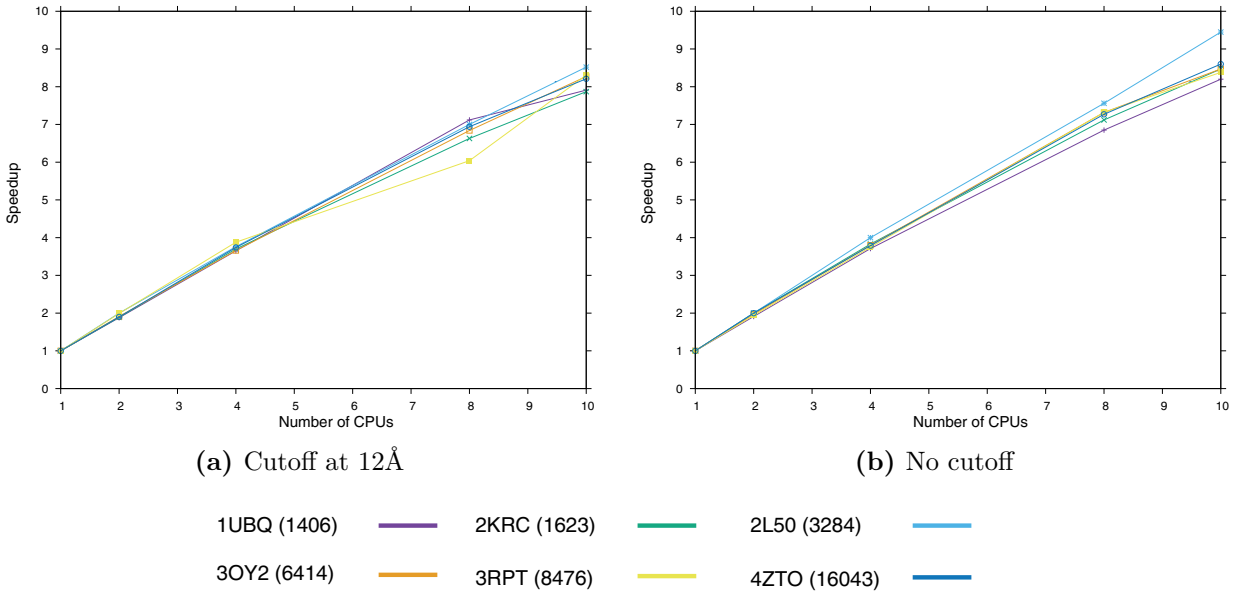
(a) Cutoff at 12Å **(b)** No cutoff

Figure 5.1.: Efficiency plots for gradient calculations. The system sizes vary from 1400 to about 16000 atoms. Figure (a) shows the results for calculations with a 12Å cutoff radius. Figure (b) shows the results for the same calculations without using a cutoff radius.

The efficiency was calculated with respect to a run on a single processor. The efficiency can be calculated as follows

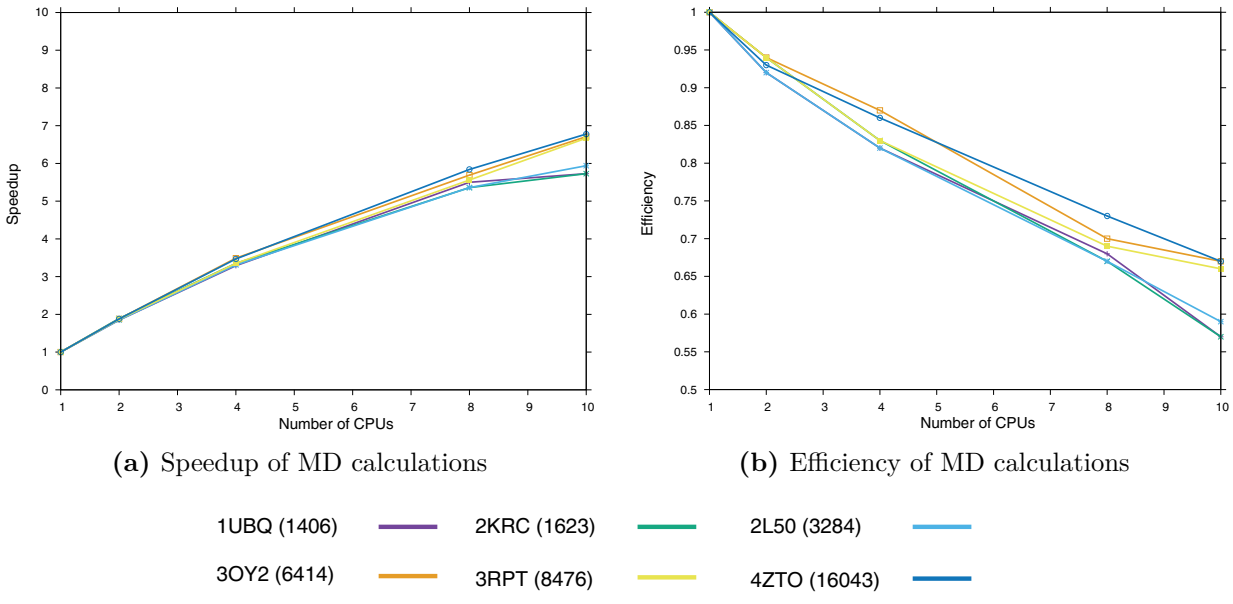
$$Eff = \frac{time_{singeCPU}}{time_{multipleCPU} \cdot N_{CPU}} \tag{5.1}$$

A perfect scaling would therefore result in an efficiency of 1, regardless of the number of CPU's. The results, depicted in Figure 5.1, clearly show that even partial parallelization of the force fields increases the performance significantly. The speedup as well as the efficiency is directly linked to the system size. For the calculations with the 12Å cutoff even the smallest system (1UBQ) shows a good efficiency of over 80% up to 10 CPU's. If the system size is increased, the changes in efficiency are noticeable but not very significant. All systems are very close in efficiency with a value of about 0.8 - 0.85. The drop in efficiency for the 3RPT at 8 CPUs may be due to other processes running on the node during parallel execution.
For the calculations without a cutoff radius the picture doesn't change very much. All systems show a great efficiency up to 10 CPUs with the 2L50 at the maximum with almost 0.95. The general trend compared to Figure 5.1 (a) is a minor increase in efficiency for all systems. Next to the efficiency the net speedup for increasing numbers of processors is shown in Figure 5.2.



(a) Cutoff at 12Å **(b)** No cutoff

Figure 5.2.: Speedup plots for gradient calculations. The system sizes vary from 1400 to about 16000 atoms. Figure (a) shows the results for calculations with a 12Å cutoff radius. Figure (b) shows the results for the same calculations without using a cutoff radius.



(a) Speedup of MD calculations **(b)** Efficiency of MD calculations

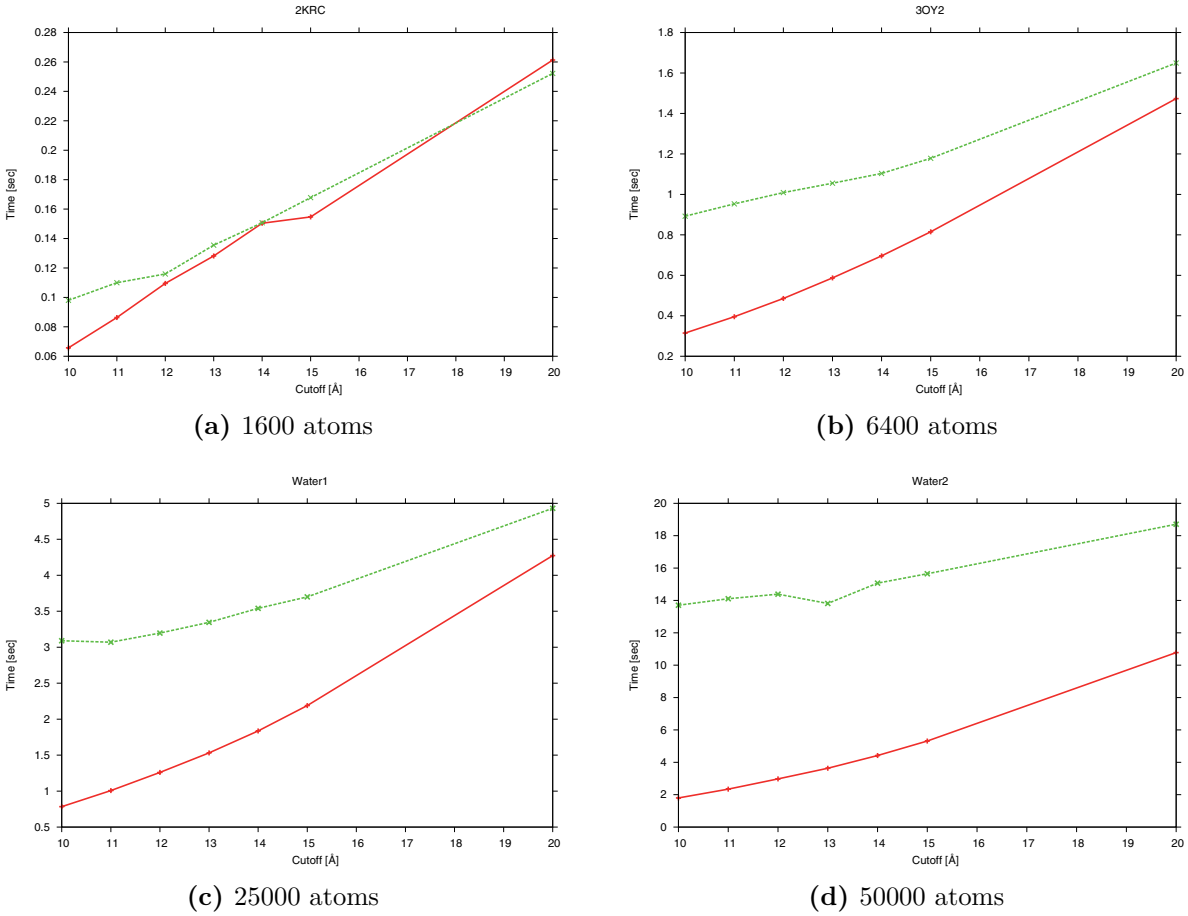
Figure 5.3.: Speedup and efficiency plots for molecular dynamics calculations. The system sizes vary from 1400 to about 16000 atoms. Figure (a) shows the results for calculations with a 12Å cutoff radius. Figure (b) shows the results for the same calculations without using a cutoff radius.

Figure 5.2 shows the speedup for the gradient calculations. For all systems, the speedup

is nearly linear up to 4CPUs. If the number of CPUs is further increased, the net speedup for each CPU shows a little drop. Nevertheless using 10 CPUs increases the speed of the calculation 7 to 8 fold for a calculation using a cutoff radius, and 8 to 9.5 fold for calculations without cutoff radius.
Since gradient calculations are the optimal case for parallel testing because no other subroutines than the parallelized ones have to be called, short molecular dynamics simulations have been performed. For each system 100 steps of MD were performed with and without using a cutoff radius. For the calculations using the cutoff radius, it was set to 12Å. The simulations were performed with the velocity Verlet integrator in the NVT ensemble using the Nóse-Hoover thermostat. Figure 5.3 shows the results for the MD simulations. Since the MD subroutine contains algorithms which are not parallelized (temperature and pressure control, constraints) or can only be parallelized with poor efficiency using OpenMP (SPME algorithm) the overall performance is not as good as for gradient calculations. Up to 8 CPUs the decrease in efficiency is similar for all systems. Starting at 8 CPUs the smaller systems up to the 2L50 show a significant drop in efficiency below 60%. The bigger systems are very close to each other with a net efficiency between 65% and 70%. The same trend can be seen in the diagram for the net speedup. Up to 4 CPUS the speedup for all systems is good. If the number of CPUs is further increased the performance drop increases. At 8 CPUs the net gain is almost 0 for the 1UBQ and only significantly better for the 2KRC and 2L50. The bigger systems still show a performance increase to up to 7 times for 10CPUs.
Summing up, the performance of CAST has been significantly increased due to the partial parallelization of the force fields. Depending on the task, the performances show an almost linear scaling with the number of CPUs even if only the non-bonded subroutines have been subjected to the parallelization.

5.1.2. Linked Cell performance

Performance of the linked-cell algorithm was tested on some of the systems used to test the OpenMP implementation. The cutoff radius was varied between 1 and 20 Å and the time for the generation of the neighbor list was measured. The generation of the main and sub-cells was neglected, since this has to be done only once at the start of the program. System sizes have been varied between 1600 and 50.000 atoms. Figure 5.4 shows the performance of the linked-cell compared to the standard Verlet list generation for four different systems. For small systems, the primitive algorithm is as fast as the linked-cell. The bigger the systems get, the better is the performance of the linked-cell.



(a) 1600 atoms

(b) 6400 atoms

(c) 25000 atoms

(d) 50000 atoms

Figure 5.4.: Performance of the linked cell algorithm (red), compared to the standard list algorithm (green) for 4 different structures. The linked cell outperforms the standard procedure for Verlet list generation especially for big systems of more than about 2000 atoms.

The standard range for cutoff radii in simulations is between 10 and 16Å. For the 2KRC the difference in computation time is almost nonexistent. The 3OY2 shows a speedup between 1.2 and 3 times depending on the cutoff radius. For the solvated system with about 25.000 atoms (Water1) the speedup ranges between 1.4 and 4.35. As expected, the biggest performance boost is achieved for the biggest system. Water2 consists of 50.000 atoms and the linked-cell generates the Verlet list about 3 to 7 times faster than the primitive algorithm.

5.1.3. Free energy algorithms

Umbrella sampling For the verification of the umbrella sampling part of CAST, two examples were chosen. First, the torsion implementation was tested by calculation of the rotational barrier of butane which is a well known literature example [228–231]. The reaction coordinate was divided into 73 windows, each covering 5°. The simulation was performed under a constant temperature of 300K with a timestep of 1fs using the CHARMM22 force field. For each window 500ps of simulation data was sampled amounting to a total of 36.5ns simulation time. The spring constant for the restraining potential of the torsion angle was set to 0.05 $kcal/(mol \cdot deg^2)$. The raw data of the 73 simulations were combined using the WHAM program. The resulting PMF is depicted in Figure 5.5.

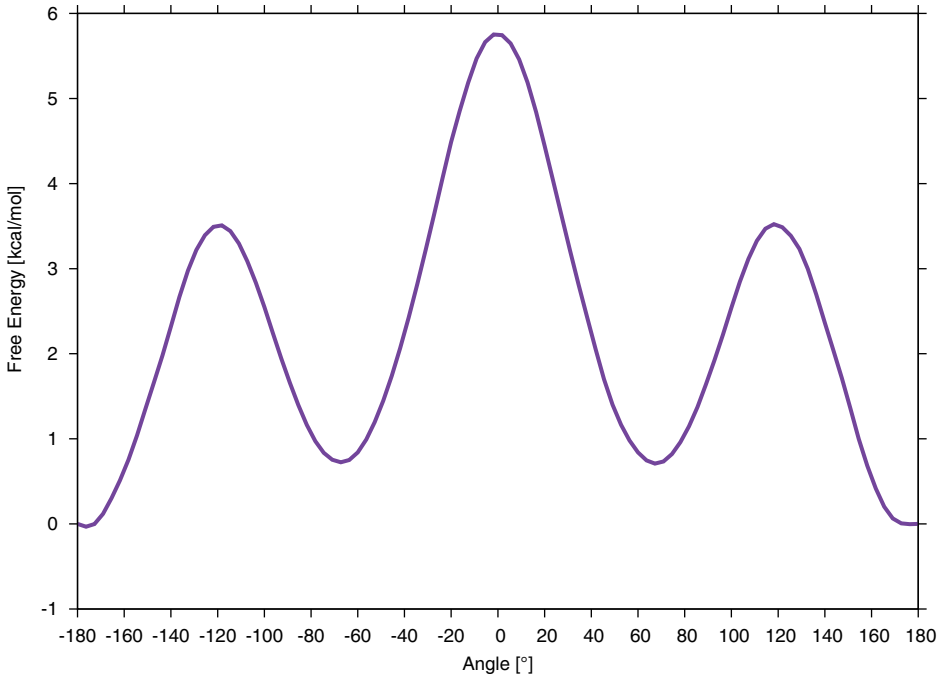


Figure 5.5.: Potential of mean force for the butane rotation. The PMF shows the significant peaks for the cis (middle) and the two gauche conformations (left, right).

The results for the rotation free energy are in almost perfect agreement with the results of Hudson using the off-path simulation (OPS) method in CHARMM [228]. The main spike in the rotation at 0 degree is recovered with about 5.76 kcal/mol. The two minor spikes, resembling the gauche conformation of the butane are also recovered nicely.
Testing of the distance restraint was performed on a small peptide. A famous example is the unfolding of deca-L-alanine in vacuum. The natural state of this peptide in vacuum is an α-helix [232]. Unfolding can be achieved by exerting a force on the first and last

α-carbon in the system. In the folded state the distance between these carbon atoms amounts to roughly 14Å. In the following calculations this distance was used as the reaction coordinate. The reaction coordinate was divided into 21 pieces, each covering a distance of 1Å and starting at a minimum distance of 12Å. Hence, the simulation covers a distance of 12 to 32Å between the two α-carbons. In order to reduce strong distortions of the highly charged carboxylate group the peptide was saturated at the carboxylate end with a NH_2 group (see Figure 5.6).

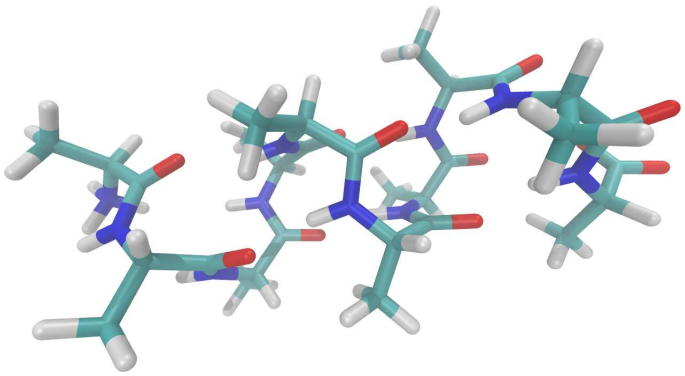


Figure 5.6.: Deca-L-alanine for the testing of the distance constraints in the umbrella sampling implementation. The carboxylate end was saturated with a methyl moiety to avoid strong fluctuations due to the negative charge of the oxygen.

All 21 simulations were performed using the CHARMM22 force field in the NVT ensemble employing a constant temperature of 300K with a 1fs timestep. For each window the system was equilibrated for 10000 steps with 500000 steps production runs. Statistics were gathered every 2 steps. The force constant for the distance restraint was set to 3 $kcal/(mol \cdot Å^2)$. The output of the umbrella sampling simulations was combined with the WHAM program and is shown in Figure 5.7. The results are in nice agreement with literature values, obtained using the ABF method [110] as well as steered molecular dynamics using Jarzynski's identity [233]. The minimum distance of about 14.5 Å, which resembles the minimum distance in an α-helix is nicely reproduced. The minimum is followed by a steep ascent between 15Å and 25Å. In this region the hydrogen bonds, stabilizing the helix are broken. The final part of the PMF corresponds to the almost fully stretched peptide where no more hydrogen bonds have to broken for further extension. The increase in free energy is therefore not as big as in the 15-25 Å region.

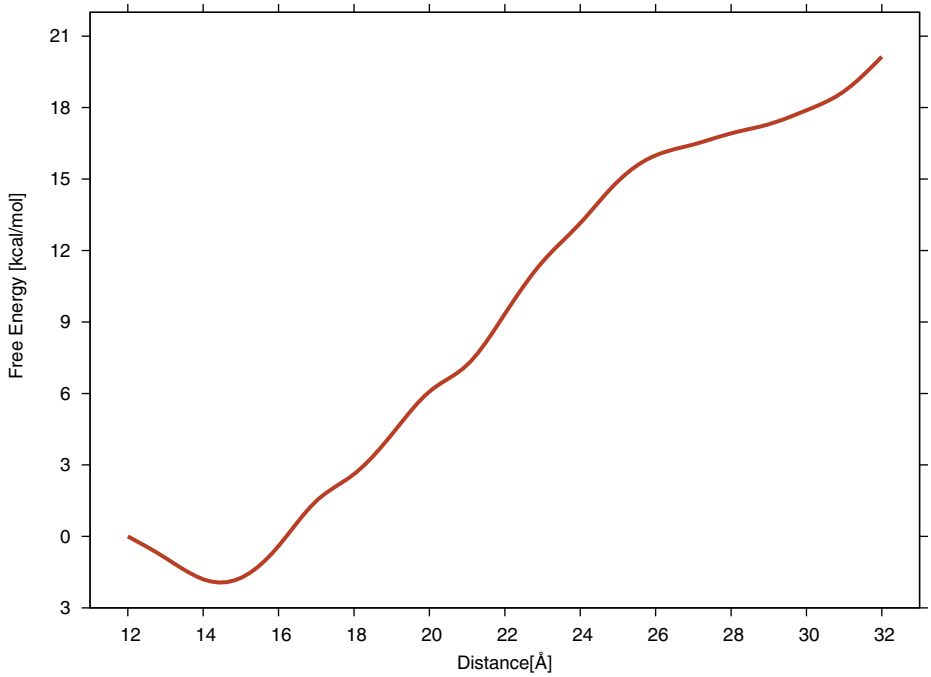


Figure 5.7.: Potential of mean force for the unfolding of deca-L-alanine in vacuum. As reaction coordinate the distance between the first and last α-carbon in the peptide was chosen. The reaction coordinate was cut in to 21 windows. Energy is given in kcal/mol

Free energy perturbation In the following section the FEP implementation is verified on three examples and compared to results obtained with NAMD. First a very simple transformation of ethane into ethane in explicit solvent is performed. The second example is the charging of a sodium ion in water. Charging free energies are well investigated in the literature, so detailed results for comparison are available. The third example is the transformation of the tripeptide ALA-TYR-ALA to a trialanine. This resembles a good model system for calculations in the most common area of use for FEP, mutations in real enzymes. All following calculations (zero-sum, ion charging and peptide mutation) were performed using the CHARMM22 force field.

Zero-sum transformation The transformation from a chemical species into the same chemical species is called a zero-sum transformation. Since this kind of transformation is symmetric, the resulting PMF is also symmetric at the midpoint and the net free energy change is expected to be zero. As a simple test-case the transformation of ethane into ethane in explicit solvent was used. The transformation was split into 20 windows, employing an increment of the order parameter λ of $\triangle\lambda$ of 0.05. Each transformation was performed under constant pressure of 1atm and temperature of 300K in a simulation

box containing 339 TIP3P water molecules. All hydrogen bonds were constraint using the RATTLE algorithm and the timestep was set to 1fs. A cutoff radius of 11 Å was applied and the long range electrostatics were handled using the particle mesh Ewald method. The simulation was repeated 3 times to get a statistical average. Before the transformations, the system was equilibrated under the above mentioned conditions for 50.000 steps (50ps). In each run equilibration was performed for 1000 steps, and production for 5000 steps in each window. A comparison with the NAMD program packaged is depicted in Figure 5.8 and Table 5.1.

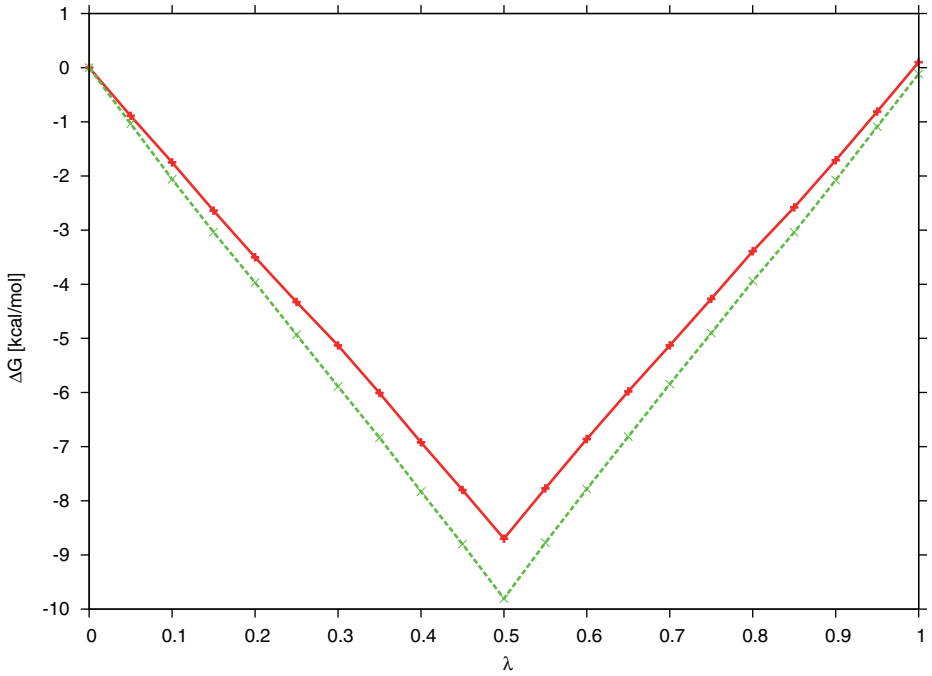


Figure 5.8.: Free energy change for the zero-sum transformation of ethane in bulk water. The green curve is the average over 3 runs calculated with NAMD. The red curve is the average over 3 runs calculated with CAST.

The results for CAST and NAMD are nearly identical. The deviation from the expected results (0.0 kcal/mol) in average is -0.1 kcal/mol for NAMD and 0.1 kcal/mol for CAST. The maximum deviation in a single run is 0.2 kcal/mol for CAST and -0.2 kcal/mol for NAMD. Also the shape of the curves shows the expected symmetry around the midpoint ($\lambda = 0.5$) of the transformation.

	CAST	NAMD
Run1	0.07	-0.239
Run2	0.22	0.046
Run3	0.02	-0.158
Average	**0.11**	**-0.117**

Table 5.1.: Results for the zero-sum transformations. Three transformations were carried out with each program to get an average. Energies are given in kcal/mol.

Ion charging Charging of ions in solution is a good example to show why the individual decoupling of charge and Lennard-Jones potential can be important. At the start of the simulation, the sodium is completely neutral but is still present in the simulation. The charge interactions with the environment are zero, but the van-der-Waals interactions are fully present. If both interactions were to be scaled with the same factor, the sodium atom would practically be not existent. In this case the water molecules could simply overlap with the sodium until the "charging" begins. This can be circumvented with the current implementation. For the complete simulation the van-der-Waals interactions are coupled with 1, employing a standard Lennard-Jones potential. Only the charge interaction is scaled during the different simulation windows while λ moves from 0 to 1. This greatly improves convergence of the simulation and eliminates possible inconsistencies when the Lennard-Jones part would have to be perturbed. The simulation was run under constant pressure (1atm) and temperature (300K) with constraint hydrogen bonds (RATTLE). The timestep was set to 1fs and the system was equilibrated for 50.000 steps prior to the FEP simulations. 3 transformations were performed with 5000 steps equilibration and 10000 steps production in each window. The reaction coordinate was separated into 20 windows with $\Delta\lambda$ of 0.05. The transformation was carried out in a box of 823 TIP3P water molecules. Results are depicted in Figure 5.9 and Table 5.2

All 3 simulations are nearly identical and converge to a value of around -89 kcal/-mol. The literature values for the charging of a sodium ion in water range between -87.2 kcal/mol obtained experimentally by Marcus [234] and theoretical investigations by Straatsma and Berendsen with a calculated free energy of -121.4 kcal/mol [235]. Another theoretical investigation was performed by Hummer [236] and the free energy change amounted to -96.8 kcal/mol. The rather large difference of the two theoretically obtained results can be attributed to two different simulation parameters. The simulation done by Hummer was perforemd in a box of 256 SPC water molecules and employing a different set of Lennard-Jones parameters for the sodium ion. Straatsma

used the standard parameters and a box of 216 SPC water molecules. Since the size of the water box should only have a minor effect on the result, compared to a different set of parameters, it is apparent that the charging free energy is very sensitive with regard to the force field parameters.

	CAST	**NAMD**
Run1	-88.58	-89.86
Run2	-88.09	-89.53
Run3	-88.69	-88.83
Average	**-88.45**	**-89.40**

Table 5.2.: Results for the charging of a sodium ion in aqueous solution. Three runs were performed with each program to get an average. Energies are given in kcal/mol.

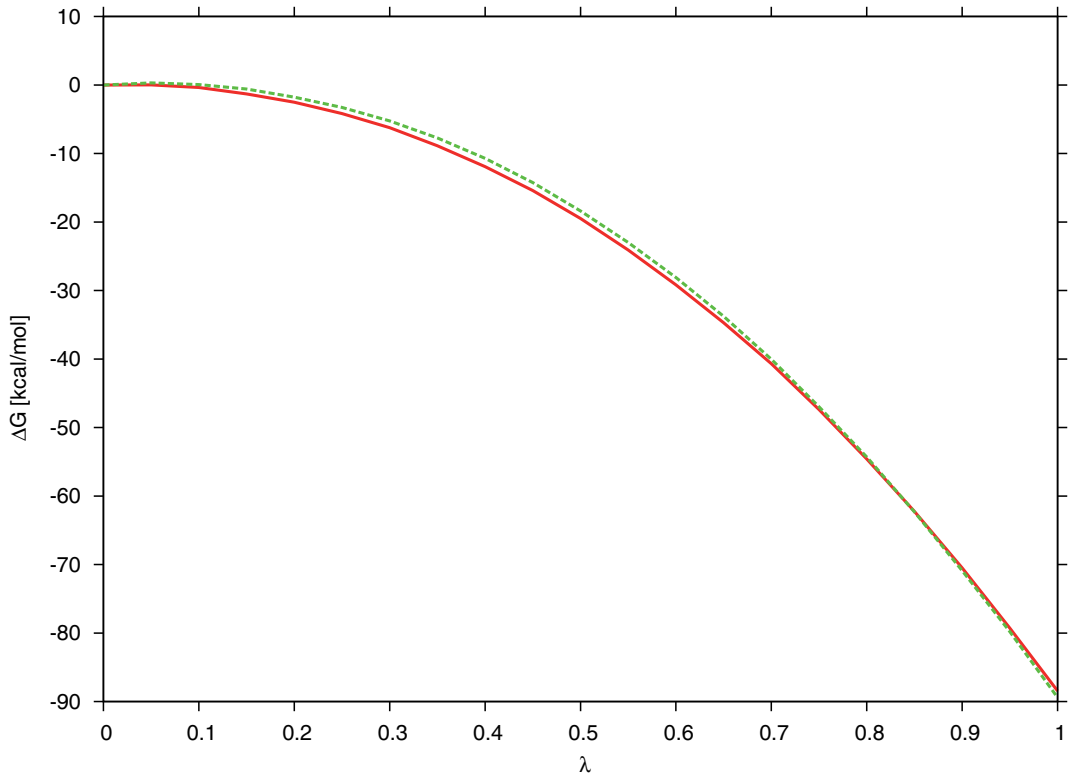


Figure 5.9.: Free energy change for the charging of a sodium ion in bulk water. At the beginning of the simulation the net charge is zero. During the simulation the charge of the sodium is increased until it reaches +1 at $\lambda = 1$. The green curve shows the average over three runs with NAMD, the red curve shows the average over three runs with CAST.

Peptide mutation One very important quantity of interest is often the free energy of hydration. It can be calculated according to the following thermodynamic cycle illustrated in Figure 5.10

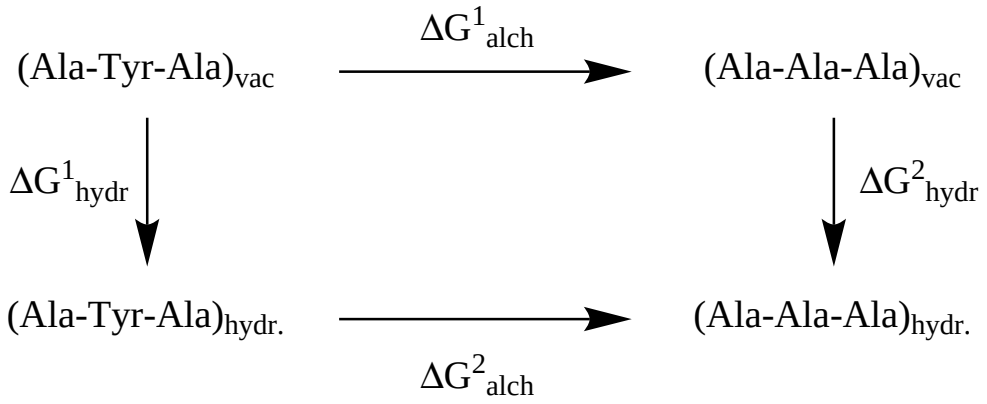


Figure 5.10.: Thermodynamic cycle for the transformation of the tyrosine into alanine. Using the differences in free energy change, the free energy of hydration can be calculated.

The vertical arrows in the cycle correspond to the hydration of the wild type tripeptide and its mutant. The horizontal arrows correspond to the respective point mutation. The relative free energy of hydration can then be calculated by

$$\triangle\triangle G = \triangle G^2_{alch} - \triangle G^1_{alch} = \triangle G^2_{hydr} - \triangle G^1_{hydr} \tag{5.2}$$

The relative free energy of hydration was calculated for a sample tripeptide with the sequence ALA-TYR-ALA in the starting state in explicit solvent with 461 TIP3P water molecules. During the transformation, the tyrosine was transformed into an alanine. To obtain the necessary data, two transformations have to be carried out, one in vacuum and one in solvent. In both cases the reaction coordinate was split into 20 windows. The vacuum structure was equilibrated for 150.000 steps prior to the alchemical transformation. The equilibration was performed under a constant temperature of 300K with a timestep of 1fs. All hydrogen bonds were constrained using the RATTLE algorithm. The cutoff radius was set to 11Å. The following transformation in vacuum was carried out three times to get a statistical average. All transformations were started from the same equilibrated structure, with 5000 steps equilibration and 40.000 steps production per window. All other simulation parameters were identical to the equilibration simulation.
The solvated structure was equilibrated for 50.000 steps prior to the transformations.

Equilibration was performed under constant temperature of 300K, constant pressure of 1atm and a timestep of 1fs. Hydrogen bonds were constrained using the RATTLE algorithm. A cutoff radius of 11Å was used. Long range electrostatics were handled using the particle mesh Ewald method. The alchemical transformations in solvent were carried out under the same conditions as the equilibration run, using 1000 steps for equilibration and 5000 steps for production in each window. Results of the calculations were again crosschecked with NAMD and are shown in Figure 5.11 and Table 5.3.

	Vacuum			**Solvent**	
	CAST	**NAMD**		**CAST**	**NAMD**
Run1	4.71	4.34	Run1	11.89	10.44
Run2	4.00	4.55	Run2	11.89	11.62
Run3	4.32	5.01	Run3	10.57	11.65
Average	**4.34**	**4.63**	**Average**	**11.45**	**11.24**

Table 5.3.: Free energies changes for the alchemical transformations in vacuum and solvent for the peptide mutation. Three runs were performed for each transformation using the same conditions for CAST and NAMD. Energies are given in kcal/mol.

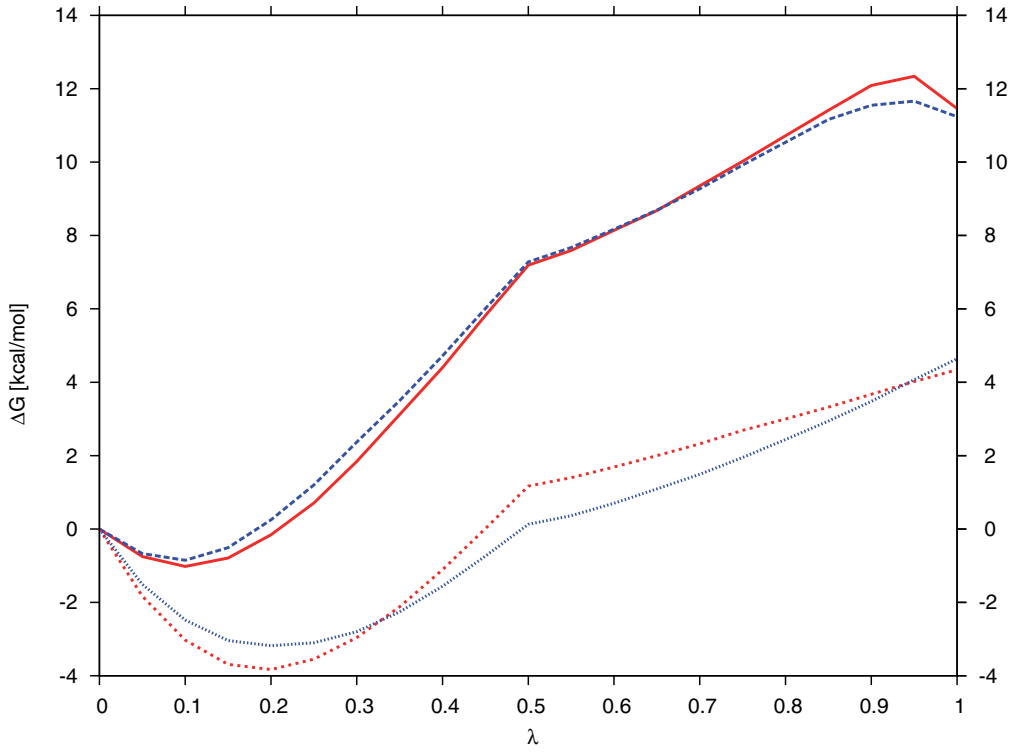


Figure 5.11.: Free energy changes for the transformation in vacuum (lower lines) and in solvent (upper lines). CAST results are shown in red, NAMD results in blue.

The transformations in vacuum and the transformations in solvent are nearly identical when comparing the NAMD and CAST results. The deviation in absolute values in vacuum is at most 0.7 kcal/mol. Furthermore, the slopes of the curves are in very good agreement as well. Looking at the simulations in solvent, the trend is the same. All simulations are also in good agreement with the NAMD reference results. The calculated net free energy change in solvent for the NAMD runs is 11.24 kcal/mol while 11.45 kcal/mol were obtained for the CAST runs. Now, the free energy of hydration can be calculated according to the thermodynamic cycle by subtracting the vacuum free energy change from the free energy change in solvent. The corresponding relative free energies of hydration are 6.61 kcal/mol (NAMD) and 7.11 kcal/mol (CAST).

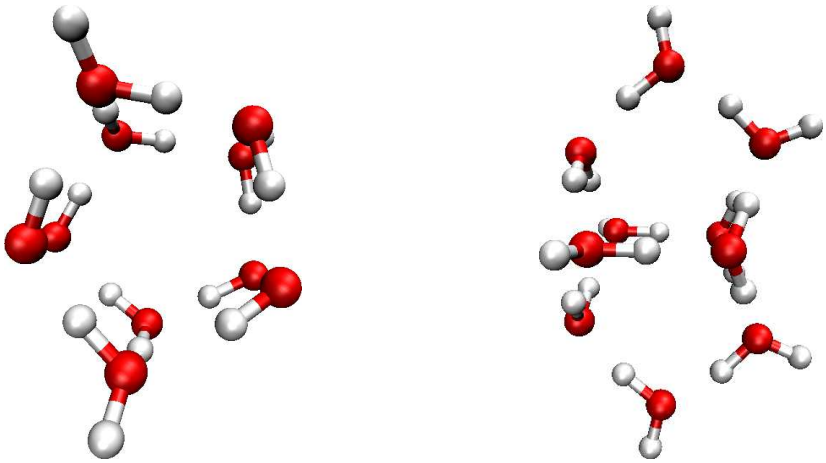
5.2. Global optimization using CAST/TeraChem

Global optimization of the $(H_2O)_{10}$ cluster Many studies have investigated the global and low lying local minima configurations of water clusters of different sizes [237–244] including the global minimum of the water decamer, $((H_2O)_{10})$ [237, 238, 243, 244]. The water decamer has therefore been chosen as a first test case for the CAST/TeraChem MPI interface. The global optimization of the decamer was performed with the TabuSearch, one time with force field methods implemented directly in CAST (OPLS-AA/TIP3P water model) and one time using DFT methods with TeraChem. Performing the global optimization with the Tabu-Search in CAST in conjunction with the TIP3P [245] water potential leads to the two lowest lying minima (see Figure 5.12) after 614 TabuSearch iterations. The energy ordering of the minima although is described incorrectly by this potential in comparison with the previous benchmark results [238, 243]. Starting the global optimization from the same initial points but using TeraChem with DFT (B3LYP-D [246–250]/6-31++G [251–259]) resulted in the putative global minimum structure found previously using higher levels of theory [238, 243]. These findings show the necessity of using QM methods to get sufficiently reliable and accurate results. The final structures are shown in Figure 5.12 and the corresponding relative energies are given in Table 5.4. The average time of one optimization cycle (local optimization + dimer search) with TeraChem amounts to 486 seconds. 43 iterations were needed to reach the global minimum. Since the communication between CAST and TeraChem is managed by the MPI interface and is performed on the same node, the communication time between the programs is quite fast and can be neglected in comparison to the required computational time for calculating energy and gradients. The results contained in reference [237] were obtained by carrying out simulated an-

nealing simulations starting from at least 4 random starting structures of the respective cluster size. The minimum structures were then optimized at the HF level using a polarized double-zeta basis. The single point energies were calculated using Møller-Plesset second-order perturbation theory (MP2).

Structure	OPLS-AA/TIP3P	B3LYP/6-31++G	MP2/6-311++G(2d,2p) [237]
Figure 5.12 (a)	0.3	0.0	0.0
Figure 5.12 (b)	0.0	1.3	2.87

Table 5.4.: Results for the global optimization using TabuSearch for a $(H_2O)_{10}$ cluster. OPLS-AA results were obtained by the force field implementation of CAST, B3LYP results are from the CAST-TeraChem interface. The global minimum for the corresponding method is always set to 0.0 kcal/mol. When comparing (a) and (b) it can be seen that the lowest minimum has changed when going from OPLS-AA to B3LYP. Energies are given in kcal/mol



(a) **(b)**

Figure 5.12.: Best two structures from of global optimizations using Tabu-Search. The energetic order differs in OPLS-AA and B3LYP; OPLS-AA: b) is the lowest minimum; B3LYP: a) is the lowest minimum

Global optimization of [Met5]enkephalin For large flexible chain molecules an exhaustive conformational search is more challenging [260–262]. Since [Met5]enkephalin was carefully investigated in previous investigations [207,263–266], we used it as a proof of principle for the general applicability of our new combination of programs. The gas phase *ab-initio* calculations are performed with the BLYP [267–269] functional in combination with the smaller 6-31G basis set [251]. Figure 5.13 shows the result of the global optimizations. Figures 5.13 (b) and 5.13 (c) give the structures obtained for the

global optimization performed with the OPLS-AA force field. Structure (b) is obtained by pure force field calculations, while (c) is the reoptimized structure (b) at the BLYP-D/6-31G level. Comparisons with the results of other investigations are difficult since they either performed simulations in solvent or did not give the full structures of the minima.

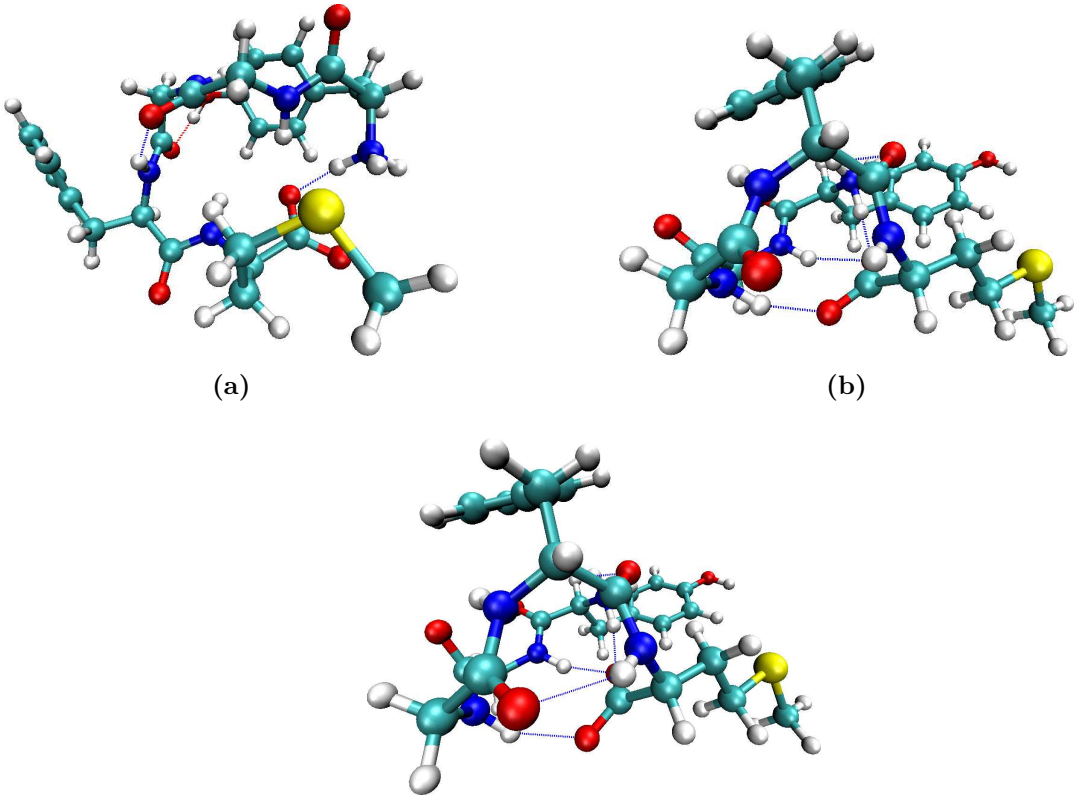


Figure 5.13.: Final structures of [Met5]enkephalin from the global optimization using (a) BLYP-D/6-31G, (b) OPLS-AA. The upper structure (b) is directly obtained from the force field search while the lower one (c) is reoptimized at the BLYP-D/6-31G level starting from the structure (b). Hydrogen bonds are indicated if the atomic distance was smaller than 1.9 Å.

Structures (b) and (c) differ considerably from structure (a) whereas the direct comparison between (b) and (c) shows no significant differences. The direct optimization with DFT yields a global minimum which is about 3 kcal/mol lower in energy than the one obtained with the force field. If the global minimum obtained with the force field is reoptimized at the BLYP-D/6-31G level the resulting structure is about 5.2 kcal/mol higher in energy than the global minimum directly obtained with the DFT methods (see Table 5.5). This clearly shows that a preoptimization with force fields in combination with reoptimization using higher level methods does not necessarily lead

to a lower minimum.

Structure	(a) BLYP/6-31G	(b) OPLS-AA	(c) OPLSA-AA + BLYP-D/6-31G
$\triangle E$	-20.3	-17.5	-15.1

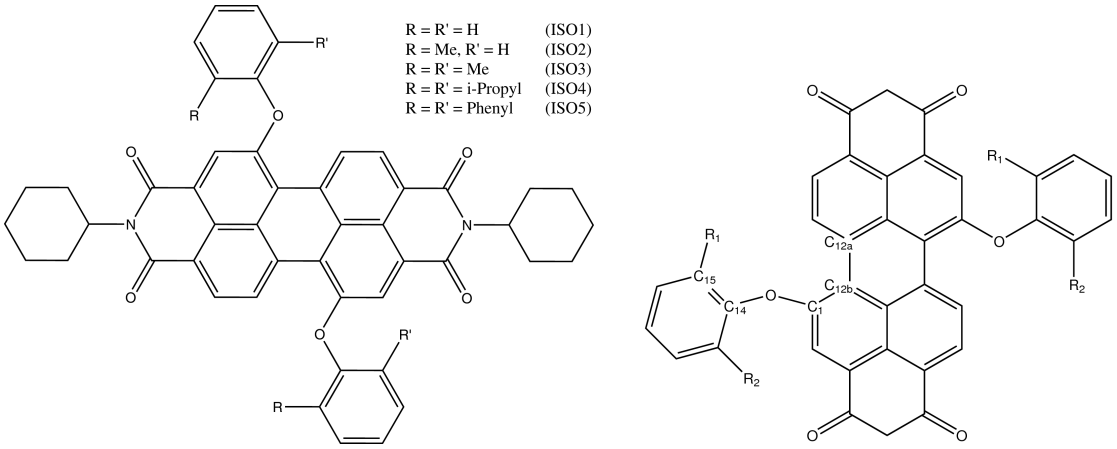
Table 5.5.: Results for the global optimization using TabuSearch for Met-Enkephalin. OPLS-AA results were obtained by the force field implementation of CAST, BLYP-D results are from the CAST-TeraChem interface. The energies given are relative energies comparing the starting structure with the global minimum found for the respective method (a,b). The energy for structure (c) is obtained by reoptimizing the OPLS-AA minimum with BLYP-D/6-31G and comparing the result with the starting structure. Energies are given in kcal/mol

5.3. Flexibility of PBI sidechains

Motivation For π-conjugated molecules, particularly polycyclic aromatic hydrocarbons, it is usually assumed that they have a planar structure [270]. A major factor is the gain of resonance energy due to the conjugated π-system and delocalized electrons. Distortions are expected if bulky substituents are applied to the core because they repel each other due to steric interactions. Further possibilities for twisted π-scaffolds may be packing effects in the crystal [271, 272]. A widely used class of those polyaromatic hydrocarbons are perylene-bisimides and their derivatives. Twisting of the π core can be observed if the PBI is substituted at the bay positions (1,6,7,12 position) [273]. Such distorted PBI were synthesized by Wang et al [274] who oligomerized 2 to 3 PBIs at the bay positions. Distortion of the planar PBI core due to ligands or packing effects can have an important effect on charge transport, fluorescence, electroluminescence and other photovoltaic properties [275–278]. Until recently no crystallograhpic data was available for 1,7-diphenoxy-substituted PBIs and the assumption was that due to the steric effects of the bay substituents a twist of 5° to 10° would be observed. Surprisingly the crystal structure of 1,7-bis(2,6-diphenylphenoxy)-PBI showed a planar π-scaffold. Several regioisomerically pure 1,7-substituted PBIs were prepared in the Würthner group to study the conformational rigidity and how much the geometry of the PBI can be influenced by packing effects. In the following theoretical investigations were carried out for a new set of 1,7-diphenoxy-substituted PBIs, synthesized by Würthner et al. Synthesis and theoretical results are published in reference [279].

PBI derivatives The PBIs used for the following studies are sketched in Figure 5.14. During the experimental investigations, it was observed that the UV-VIS spectra showed a sharpening in the vibronic progress when substituents are attached. In principle this

could arise if the functionalization restricts the available conformational space around the bay angle. In return, this would have a dramatic effect on the +M-effect of the oxygen, which is responsible for the batochromic shift in the $S_0 \rightarrow S_1$ transition. Distribution of the +M effect between the oxygen and the PBI core is strongly dependent on the conformation, as could be shown for tetraphenoxy-substituted PBIs.



(a) PBI derivatives **(b)** Inner and outer dihedral angle

Figure 5.14.: Derivatives of the synthesized 1,7-diphenoxy substituted PBIs (a) and definition of the inner (C12b-C1-O-C14) and outer (C1-O-C14-C15) dihedral angle (b).

The most rigidified molecules should therefore have the strongest +M effect and influence on the optical properties. To substantiate our view about the restriction of flexibility, we simulated variations in the flexibility for several of the compounds, namely for the first three substitution patterns in Figure 5.14 (a). To check the dependencies of the two important dihedral angles (see Figure 5.14 (b)), two-dimensional potential energy surfaces were computed. They depend on both dihedral angles, while all other parameters of the system were optimized at each point. Additionally, molecular dynamics without any restraints were performed.

Theoretical investigations All following calculations have been performed using the CAST program in conjunction with the MOPAC interface and the PM6-DH2 semi-empirical method. To speed up the calculations the cyclohexane rings at both ends of the PBI were removed. In a first step the size of the rotational barrier for the rotation of the sidechain around the inner dihedral angle (C12b-C1-O-C14) was calculated. For this a relaxed scan for the corresponding angle was performed. The angle was kept fixed, and the rest of the molecule was allowed to relax during the optimization. The rotation was performed from 180° to -30° in 20° steps. Definition of the angle can be

seen in Figure 5.14:
All input files were prepared manually. The input structure was defined as a MOPAC z-matrix in internal coordinates. For each step in the scan a partial optimization was performed. The dihedral angle under investigation was kept fixed and the rest of the molecule was allowed to relax. The possible different orientations of the sidechains were not taken into account, the calculations were performed to get a rough estimate of the rotational barrier. Figure 5.15 shows the results for the rotation around the C12b-C1-O-C14 angle. As expected, the barrier of the rotation around the inner dihedral increases with the size of the substituents. For ISO1 the curve is rather smooth with steady increase and the barrier at 0° is about 25 kcal/mol. For the ISO2, ISO3 and ISO5 isomers the curves exert a maximum at about 90° with small decrease in energy till 70°. The maximum for all three cases is still 0° with a barrier height of about 30 kcal/mol. The ISO4 isomer shows an alternating curve. This may be due to the fact that the i-Propyl group is in total far more bulky than the small Met or flat Phenyl substituents. Since only one angle is fixed during the relaxed scan the group is allowed to rotate during optimization and the conformation of the group itself has not to be identical for all inner dihedral values along the curve. This also explains the maximum at about 120° since the fixation of the inner angle leaves almost no space for the i-Propyl group.

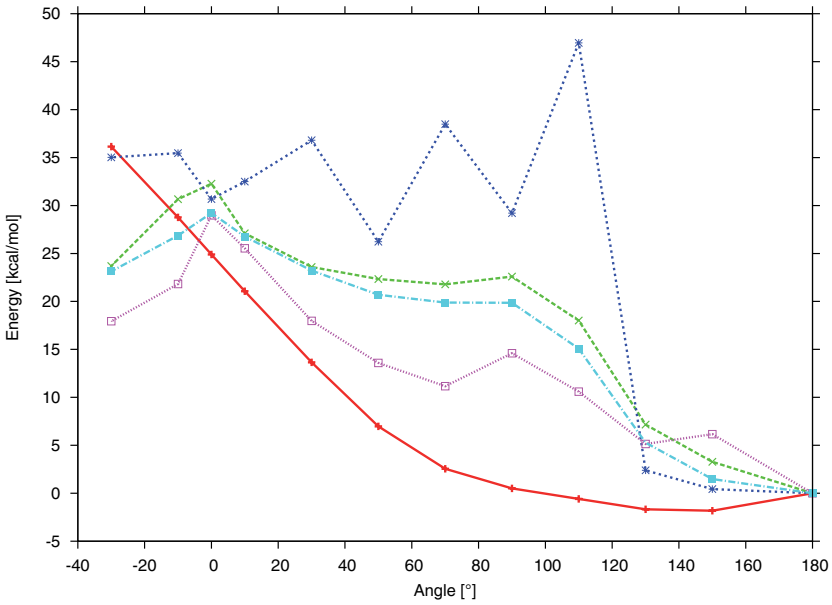
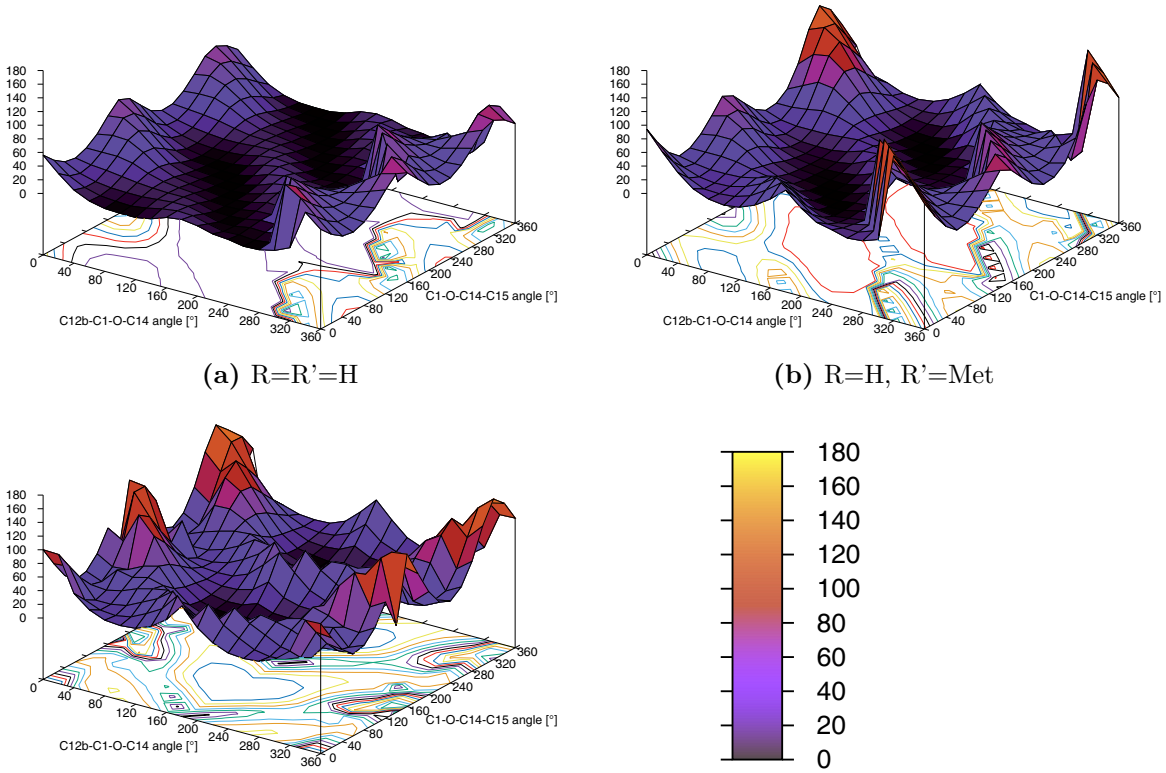


Figure 5.15.: Energy profile for the rotation of the PBI sidechain around the inner dihedral angle. Calculations were performed for R = R'=H (red), R = H, R' = Met (purple), R = R' = Met (cyan) R = R' = iProp (green) and R = R' = Phenyl (blue) substituiton patterns.

To gain a deeper insight into the barriers, potential energy surfaces were generated for the ISO1, ISO2 and ISO3 compounds, since between these compounds the largest deviations in the absorption spectra are found. The PES depend on the inner (C12b-C1-O-C14) and outer (C1-O-C14-C15) dihedral angles. All calculations were again performed with the PM6-DH2 semi-empirical method. During optimization the respective angles were kept fixed, and the rest of the system was allowed to relax. The input files were prepared accordingly to the procedure used for the rotation profile with the addition that the second dihedral angle was also kept fixed during optimization.



(a) R=R'=H

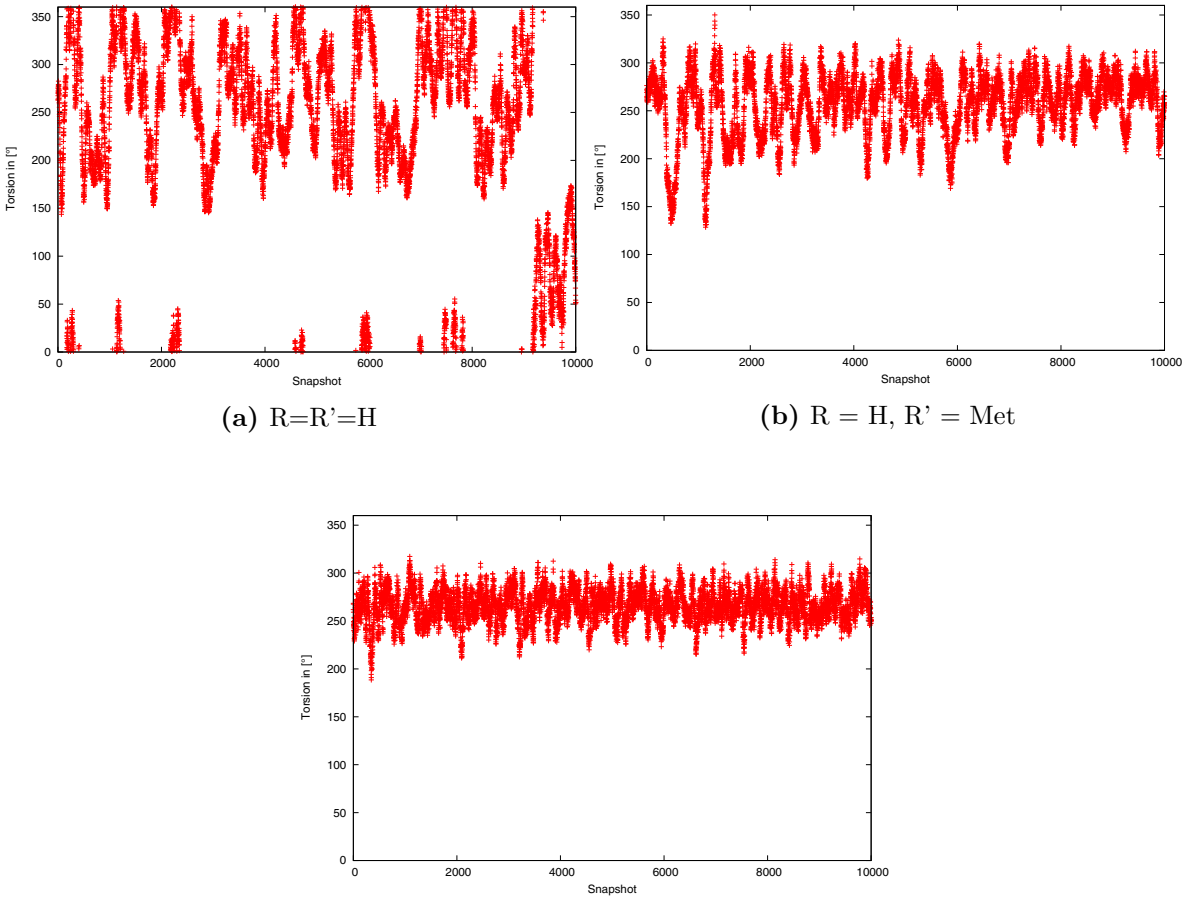
(b) R=H, R'=Met

(c) R=R'=Met

Figure 5.16.: Potential energy surfaces for the compounds ISO1 (a), ISO2 (b), and ISO3 (c). The smaller the substituent, the smaller is the barrier between the conformational isomers. The highest barrier is found for compound ISO3 which corresponds to two methyl group subsitutents in position R_1 and R_2. The color code for the energy is shown on the bottom right. Energies are given in kJ/mol.

For a more detailed description, the increments for the angle differences between two points of the surface were set to 10°. The potential energy surfaces were calculated for 180 to 0° in two dimensions, amounting to a total of 324 optimizations per surface. Since the molecule possesses symmetry, the surfaces can be extended to a full 360° rotation

by mirroring the calculated surface. The calculated surfaces can be seen in Figure 5.16. The computed two-dimensional potential energy surfaces of the three systems agree qualitatively. All exhibit two minima running parallel to each other. The orientation of both minima with respect to (C12b-C1-O-C14) and (C1-O-C14-C15) reflects the strong correlation between both torsional motions. Two minima (black areas) arise since a given substituted phenyl group can perform its torsion/rotation motion in two orientations. They lead to nearly identical energies since the relative distances to other atoms are nearly the same. However, the decreased flexibility in the series ISO1, ISO2, and ISO3 is reflected in the size of the black area of the minima which strongly shrinks along the series. For compound ISO1 (Fig. 5.16 (a)) both minima extend by about 280° for (C12b-C1-O-C14) and about 260° for (C1-O-C14-C15) (please note the periodic nature of the potential). For ISO2 the sizes of the minima shrink to approximately 220° and 200° for (C12b-C1-O-C14) and (C1-O-C14-C15), respectively, while for ISO3 a further restriction could be seen. Based on the results from the PES calculations, the energetics should allow the rotation around one of the angles for the R = R' = H and possibly R = R' = Met substituents. In order to further verify the results molecular dynamics simulations with no restrictions on atom movement were performed. For all MD simulations the CAST/MOPAC interface was used, employing the PM6-DH2 method. The input files were converted from a MOPAC z-matrix to a *xyz* file in CAST readable format using Molden. All simulations were performed over 100.000 steps with a time step of 1fs in vacuum. A constant temperature of 300K was employed using the Nòse-Hoover thermostat. Out of the 100.000 steps of the simulation, the first 20.000 steps were used for equilibration. The values of the (C1-O-C14-C15) for each 10th time step are shown in Figure 5.17 for compounds ISO1 (a), ISO2 (b), and ISO3 (c). The values for the dihedral angles have been converted to a 0° to 360° pattern. For space reasons only the values for one of the sidechains are shown. The fluctuations for the outer dihedral angle show a strong dependance on the size of the substituents. For the hydrogen substituents a rotation around the outer angle can be observed between timesteps 90.000 and 100.000. For compound ISO2 the fluctuations of the angle are already restricted to 150° - 300°. If the size of the substituents is further increased, the angle only fluctuates about 200° to 300° as seen for compound ISO3. The trend is continued for the other two PBI with even bigger substituents. A combination of the results for the computed potential energy surfaces and the dihedral angle distribution taken form the molecular dynamics simulations is illustrated in Figure 5.18.



(a) R=R'=H

(b) R = H, R' = Met

(c) R = R' = Met

Figure 5.17.: Values for the outer dihedral angle for three different substitution patterns. The fluctuation of the angle values are strongly dependent on the bulkiness of the substituents. The size of the substituents increase from (a) to (b) to (c). Only for the R = R' = H substitution pattern (a) a rotation around the angle can be observed between time steps 80000 and 100000 whereas for (b) and (c) the angle only adopts values in a range between 150°and 300°.

The surfaces are shown as a contour map and the green dots indicate the values for the inner and outer dihedral angle at a given time step. For ISO1 (Figure 5.18 (upper)) no rotation around the (C12b-C1-O-C14) (around the complete phenoxy group) can be found but as indicated by the dots in both minima, a rotation around (C1-O-C14-C15) (around the phenyl group) takes place during the simulation. However, we only observed one complete rotation in the whole simulation, indicating that this rotation is strongly restricted. As indicated by the green dots (Figure 5.18 (middle)) for ISO2, no rotation takes place and the motion is already restricted to (C1-O-C14-C15) with values in the range between 170° and 320°. For ISO3 (Figure 5.18 lower) the range further shrinks to 220° - 310°.

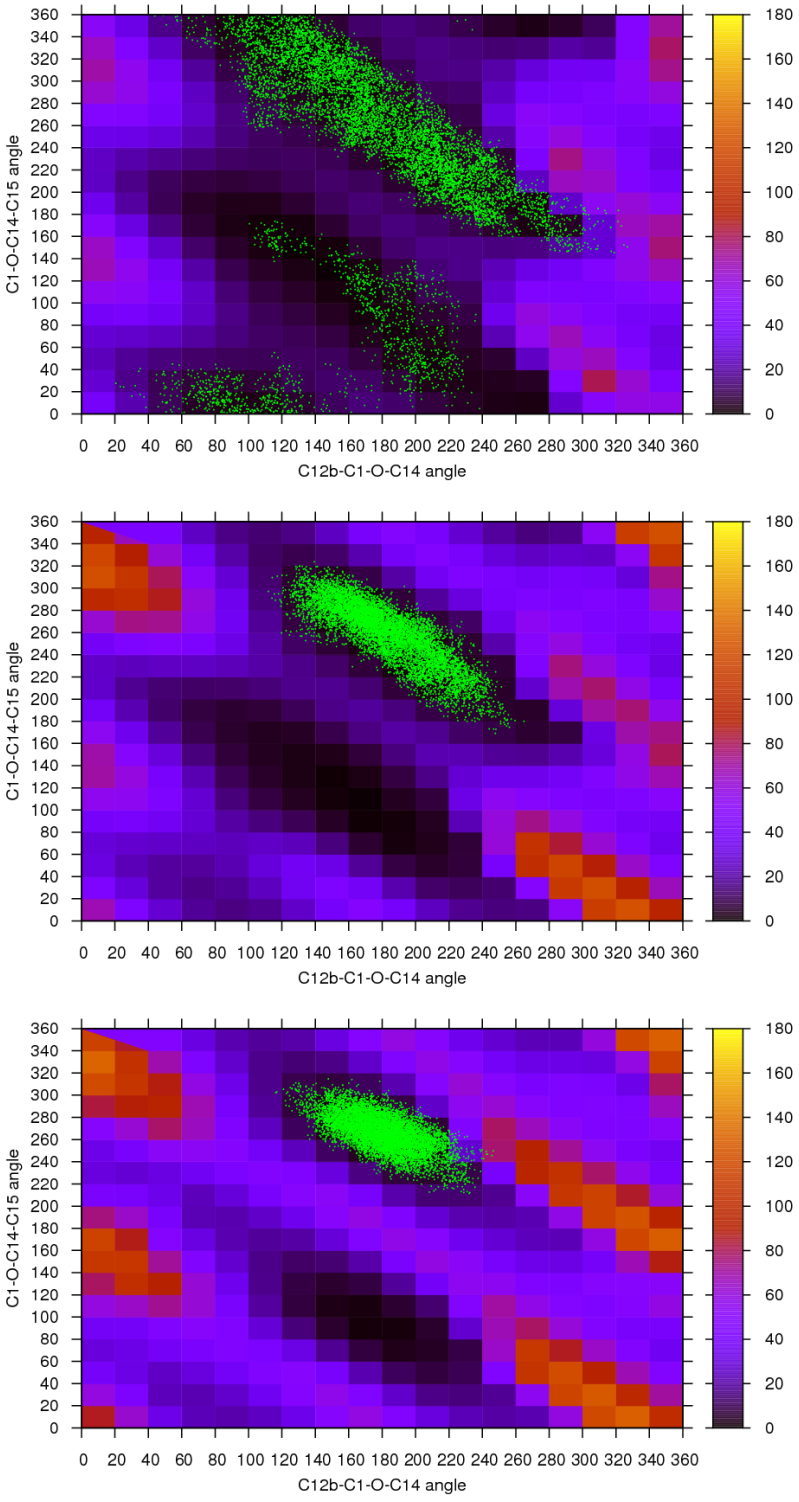


Figure 5.18.: Potential energy surfaces as a function of the inner and outer dihedral angles for compunds ISO1 (upper) ISO2, (middle) and ISO3 (lower). The green dots mark the values for the inner and outer dihedral for each 10th step of the respective MD simulation Angles are given in (°), the color code represents the energy in kJ/mol.

5.4. Relative binding free energies

Motivation The African trypanosomiasis or sleeping sickness caused by the Trypanosoma brucei is one of the most significant parasitic diseases in sub-Saharan Africa. Currently available therapies face an increasing problem of toxicity and resistance which makes the development of new drugs a significant task [280–282] for the fight against this disease. Drug targets are often proteases which are essential for the life cycle of the parasite. The trypanosoma brucei offers some potential target enzymes especially cystein proteases [283–285] of the papain superfamily. These proteases have become the main target for the development of drugs over the past years. Current inhibitors for Rhodesain of the T. bruzei rhodesiense have shown to exert considerable antitrypanosomal activity [286, 287]. For cystein proteases vinyl sulfone based inhibitors have been reported to be very effective [288–292]. They show a significant selectivity for different cystein proteases over serine proteases, good inertness in solution, if it enters the active site and a significant inactivation of the enzyme. The crystal structure of Rhodesain has first been reported with the K11777 inhibitor present as depicted in Figure 5.19.

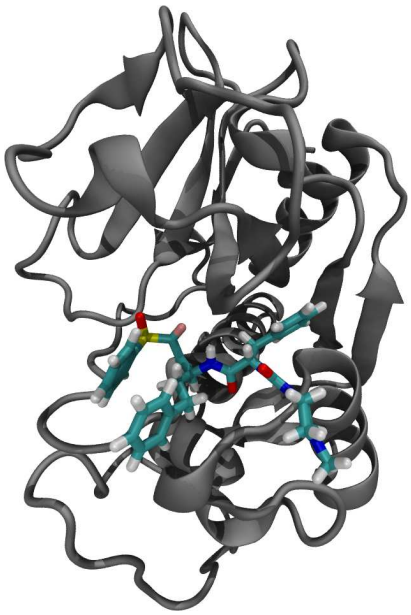


Figure 5.19.: Crystal structure of Rhodesain with the inhibitor K11777 (PDB code: 2P7U)

Inhibitors can generally be classified into two different groups, irreversible and reversible inhibitors. Irreversible inhibition usually involves the formation of a strong chemical bond between inhibitor and protein. Reversible inhibition is mostly based on non-covalent interactions between the drug and active site. However, recently more reversible inhibitors were developed, which act reversibly although a covalent bond is

formed. In such cases the reversibility results because the covalent bond is rather weak. Because irreversible covalently bound inhibitors have various disadvantages (allergic reactions etc.) [293–296], most drugs are designed to react non-covalently. Due to the weak ligand-target interactions of reversible inhibitors however, the ability to block and the residence time may be poor. The solution is to design covalent-reversible inhibitors to get the advantages (strong bonding but low to none side effects) of both types. If a covalent bond is formed, the inhibition process can be divided into two parts, the formation of the non-covalent protein-inhibitor complex and the chemical reaction of the drug with the protein (see Figure 5.20) leading to the covalent protein-inhibitor complex.

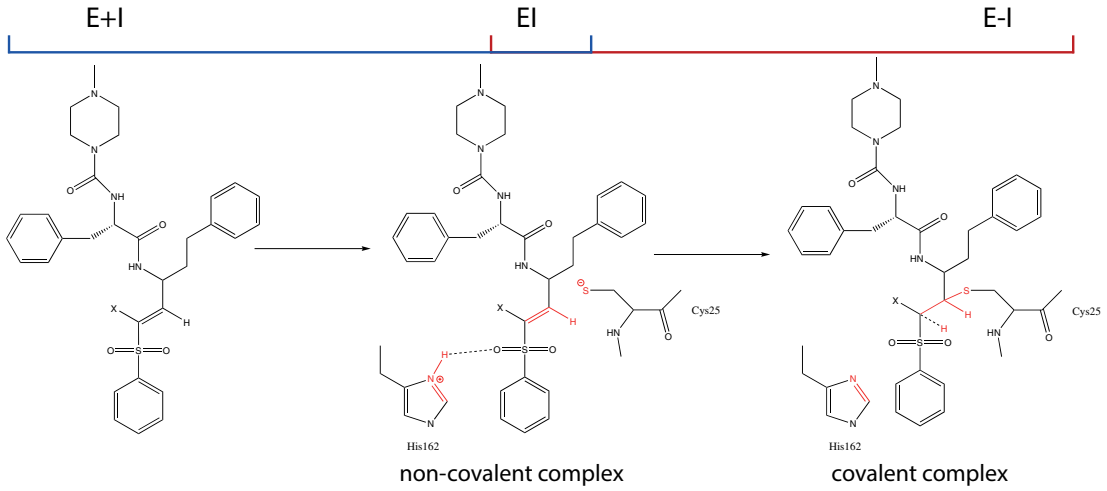


Figure 5.20.: General two step inhibition mechanism of a inhibitor. In the first step the protein-inhibitor complex is formed. After the formation, the chemical reaction takes place between the drug and the residues in the protein active site. $X = H$ (K11777): irreversible inhibition, $X = Hal$: reversible inhibition.

Engels and Schirmeister devised a protocol to design such reversible-covalent inhibitors by altering the substitution pattern in specific parts of the inhibitor and thereby altering the reaction profile of the second step as seen in Figure 5.20. Reactivity was altered by substituting the hydrogen on the double bond by halogen derivatives (F, Br). The thermodynamics and energetics of these new inhibitors going from the covalently bound to the non-covalently bound state, were examined by A. Weickert in the Engels group. While the differences in the second step are taken into account in the protocol, the question remains if and how the variation of the warhead has influence on the overall stability of the initially formed non-covalent complex. The K11777 inhibitor is known to react with the active site of the enzyme covalently. Since a covalent reaction implies a certain residual time in the active site, the non-covalent inhibitor-protein complex

for the K11777 has to be stable. Designing new inhibitors, for example by substitution patterns, therefore necessitates a knowledge about the stability of this non-covalent complex. If, for example, the hydrogen at position X (see Figure 5.20) is substituted by halogen atoms, a perturbation is introduced compared to the original inhibitor. If the non-covalent complex of this new inhibitor and the protein is strongly destabilized, a further reaction with the protein can become unlikely. By substituting the hydrogen with a fluorine chemical intuition suggests a rather small change since the radii of both atoms are nearly identical. For the change $H \rightarrow Br$ however a much larger change is expected especially due to steric reasons since the bromine possesses a much larger van-der-Waals radius. Surprisingly, in the experiments, the inhibitor substituted with fluorine leaves the active site much faster than the one substituted with bromine. The following computations investigate if this behaviour is reflected in the relative stability of the non-covalent protein-inhibitor complexes for the K11777 and two new inhibitors. Since the process of non-covalent binding usually involves entropic effects, the main driving force and quality of the inhibitor can only be measured by the free energy differences. To compute these free energy changes, we used the alchemical FEP method implemented in the CAST and NAMD program packages in conjunction with the CHARMM22 force field.

5.4.1. Technical details about the simulations

Vacuum and solvent simulations were performed with CAST and NAMD. Due to performance reasons, the simulations involving the whole protein were performed with NAMD but can in principle also be performed with CAST. Due to consistency reasons, only results obtained with NAMD are discussed in the following paragraphs. The CAST results for vacuum and solvent transformations can be found in the Appendix. The inhibitor structure was prepared by cutting the inhibitor from the *pdb* file. Parametrization was performed with the SwissParam web-server for the substituents H, F, and Br at the X position (see Figure 5.20). The protein structure was taken from the protein data base (2P7U). The resulting files were converted with the PDB converter (see Section 4.5) to yield the final structures and parameter files. For the solvent calculations the vacuum structure was solvated using the SOLVADD algorithm of CAST. According to earlier studies, the non-covalent inhibitor-protein complex in cystein proteases is formed in the charged state of the catalytic dyad [297], meaning a negative charge on the cystein (Cys25) and a protonated histidin (His162). The protein structure without the inhibitor was prepared with VMD and the CHARMM22 parameters for the protonated histidin and deprotonated cystein. The *autopsf* feature of VMD was used to add the missing

hydrogen atoms and prepare the topology of the system. The *solvate* script, also part of VMD was used to generate the water shell around the system. For the sole inhibitor the solvation shell was set to 40Å, for the inhibitor-enzyme systems the solvation shell was increased to 70Å.
NAMD employs the dual topology paradigm for the topology if FEP calculations are performed. As described in Section 4.4.2, in the dual topology both the initial and target state are present at the same time. To prepare the *pdb* and *psf* files for the FEP runs, the *alchemify* program, provided with the NAMD package was used to eliminate all possible interactions between the atoms specific to the initial and targets state. For the CAST vacuum and solvent transformations this procedure was performed with the CUT and ADD programs (see Section 4.4.2). Prior to the FEP runs, the systems containing the solvated inhibitor were equilibrated for 2ns under constant temperature (300K) and pressure (1atm). The timestep was set to 1fs. The enzyme-inhibitor complex was equilibrated under the same conditions for 5ns. Full scale electrostatics were used by imposing a smooth particle mesh Ewald method. To avoid the end point catastrophes a soft-core potential was employed for the van-der-Waals interactions. The electrostatic interactions were scaled linearly with the order parameter in a way that electrostatics of the target system started to develop beginning at the half point ($\lambda = 0.5$) of the transformation. At the same point the electrostatic interactions of the initial state were reduced to zero.

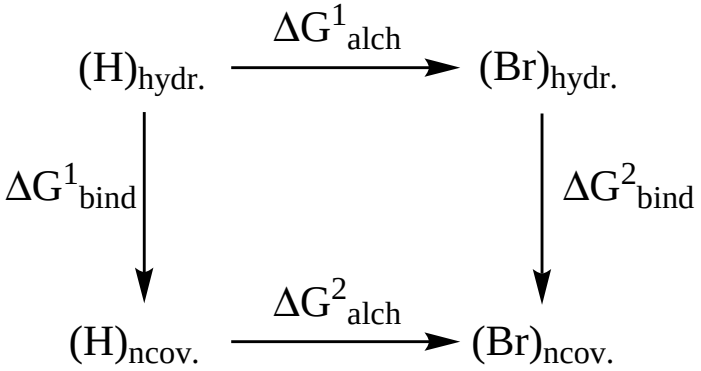


Figure 5.21.: Thermodynamic cycle for the determination of protein-ligand relative binding free energies. Instead of carrying out the extremely difficult transformations corresponding to the vertical arrows going diretly from the hydrated (hydr) to non-covalent (ncov) state, the transformations corresponding to the horizontal arrows are used. This yields the difference in binding free energy: $\Delta G^2_{alch} - \Delta G^1_{alch} = \Delta G^2_{bind} - \Delta G^1_{bind}$

The thermodynamic cycle shown in Figure 5.21 can be used to calculate the relative

binding affinity of the two inhibitors. The direct computation of the binding affinity is extremely difficult because one has to model the association of the ligand to the receptor. Due to the rearrangements in the receptor, which can be pretty large, convergence is extremely difficult to reach. A shortcut to the desired results can be taken if one is only interested in the $\Delta\Delta G$. For such a thermodynamic cycle the equation $\Delta G^1_{alch}+\Delta G^2_{bind} = \Delta G^2_{alch}+\Delta G^1_{bind}$ holds. Instead of directly calculating the more difficult $\Delta G^1_{bind}-\Delta G^2_{bind}$ one can refer to the calculation of $\Delta G^1_{alch}-\Delta G^2_{alcch}$. This is equal to the transformation of inhibitor 1 to inhibitor 2 in water and in the protein environment respectively.

Vacuum transformations

Prior to the solvent and protein transformations, vacuum transformations were carried out for the transformations $H \longleftrightarrow Br$, $H \longleftrightarrow F$ and $Br \longleftrightarrow F$. The simulations were carried out under constant temperature of 300K with a time step of 1fs. The simulation was cut into 100 windows of equal width, amounting for a $\triangle\lambda$ of 0.01. For each window, 50.000 steps equilibration and 150.000 steps production were performed. The calculation was done a total of 3 times to get a statistical average. All 3 simulations were performed bidirectional with a simulation time of 20ns for each direction. The total simulation time per run amounts therefore to 40ns.

Solvent transformation

The solvent transformations were carried out in explicit solvent in a simulation box of 38Å size including the inhibitor and 1923 TIP3P water molecules. All calculations were performed with a time step of 1fs under constant temperature (300K) and pressure (1atm). Electrostatics were treated by a particle mesh Ewald summation. All other simulation parameters are identical to the vacuum calculations.

Protein transformation

The protein transformation were also carried out in explicit solvent in a simulation box of 70Å size including the inhibitor and 7409 TIP3P water molecules. All other simulation parameters were identical to the transformations in solvent.

5.4.2. Results and discussion

Convergence of the simulations

Like with most free energy methods, FEP suffers from convergence problems [298, 299]. The average of the free energy is dominated by rare events, which manifest as abrupt changes in the free energy. This problem arises from the overlap of the equilibrium distributions $P_0(\triangle U)$ and $P_1(\triangle U)$. The main FEP equation (see also Section 3.4.2)

$$exp(-\beta\triangle A) = \langle exp(-\beta\triangle U)\rangle_0 \tag{5.3}$$

implies that the most dominant contribution to the average is based on microstates from ensemble 0, the reference ensemble, which also are microstates of the target ensemble. A small overlap of $P_0(\triangle U)$ and $P_1(\triangle U)$ implies that the important microstates are only sampled poorly which results in a poor convergence. Hence the overlap of the probability distributions is a good measure for the convergence of the simulation.
Another possibility in a case like the one presented here, is to calculate a closed thermodynamic cycle between the different transformations. In our case the inhibitors of interest posses a hydrogen or fluorine, respectively bromine atom at a specific position. The transformations of interest are therefore $H \rightarrow Br$, $H \rightarrow F$ or respectively $Br \rightarrow F$. The results for these transformations can be combined and the cycle depicted in Figure 5.22 can be calculated. Going through the cycle one time from H to Br to F to H should yield a net free energy change of zero.

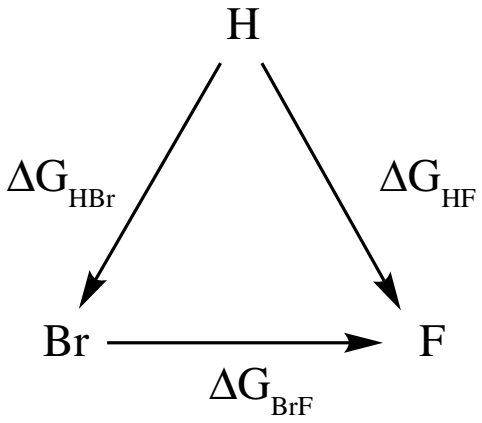


Figure 5.22.: Closed thermodynamic cycle for the transformations of the different inhibitors into another. For a converged result $\Delta G_{HBr} + \Delta G_{BrF} - \Delta G_{HF}$ is zero.

Another widely used possibility to increase accuracy and check convergence is to perform the calculations bidirectional [298, 299]. In this case results from the forward and backward transformations can be combined with the simple overlap sampling (SOS) [300] or the Bennett acceptance-ratio(BAR) [70, 301] estimator. By combining the results of forward and backward transformations the variance is reduced and the accuracy of the result is improved. Bennets acceptance ratio is defined by the following relation

$$\sum_{i=1}^{N_f} \frac{1}{1 + \frac{N_f}{N_b} e^{\beta(\triangle U_i - \triangle A)}} = \sum_{i=1}^{N_b} \frac{1}{1 + \frac{N_b}{N_f} e^{\beta(\triangle U_j + \triangle A)}} \tag{5.4}$$

with N_f and N_b as the number of samples at each state, $\beta = 1/k_B T$, $\triangle A$ as the free energy difference and U as the potential energy of steates i and j respectively. Equation

5.4 is solved numerically, normally by a Newton-Raphson [302–304] solver.
All simulations described in Section 5.4.1 were evaluated using the BAR method as implemented in the *FEPParse* plugin [298] in VMD to obtain the free energy differences. Furthermore the overlap of the underlying probability distributions was calculated.

Vacuum transformations

The transformations in vacuum show equal behavior for all three transformations (see Table 5.6). In all cases the maximum deviation between the different runs is below 1 kcal/mol. The transformation of $Br \longleftrightarrow F$ shows a positive free energy change of about 2.3 kcal/mol. This result is also recovered when the $H \longleftrightarrow Br$ and $H \longleftrightarrow F$ transformations are performed. The net free energy change for the cycle, starting from hydrogen and transforming to bromine, to fluorine and back to hydrogen is 0.11 ± 0.55 kcal/mol which is very close to the ideal result of zero.

	H⟷Br	H⟷F	Br⟷F	Cycle
Run1	26.95	29.52	2.11	-0.46
Run2	26.71	28.88	2.11	-0.06
Run3	27.07	28.84	2.63	0.86
Average	**26.91 ± 0.15**	**29.08 ± 0.31**	**2.28 ± 0.25**	**0.11 ± 0.55**

Table 5.6.: Simulation results for the single bidirectional (⟷) transformations in vacuum and averages of the three runs. The transformation of $H \longleftrightarrow F$ yields a slightly higher free energy change than for the $H \longleftrightarrow Br$ transformation. The difference is recovered for the $Br \longleftrightarrow F$ transformation. Energies are given in kcal/mol.

The results for the vacuum transformations with CAST are depicted in Table A.1. As can be seen, the results are identical to the NAMD results. The closed cycle is calculated to 0.33 ± 0.57 kcal/mol and is very close to the ideal result of zero. The absolute values for the transformations are also in superb agreement with the NAMD results.

Solvent transformations

Starting with the solvent transformations, the overlap of the underlying probability distributions was calculated to check convergence of the calculation. For all transformations performed in solvent, the overlap of the probability distributions is very good. As seen in Figure 5.23 the overlap for the transformations starting from the hydrogen are not as good as the overlap for the $F \longleftrightarrow Br$ transformation. For both $H \longleftrightarrow Br$ and $H \longleftrightarrow F$ the overlap fluctuates around 90% with drops for some windows down to 50%. The large drops however only occur in one out of the three transformations. By

comparison the overlap during the $F \longleftrightarrow Br$ transformation fluctuates around 90% with the smallest overlap for a single window still above 70%. Nevertheless convergence is still very good for all batches, and shows that the results for the free energy change are converged.

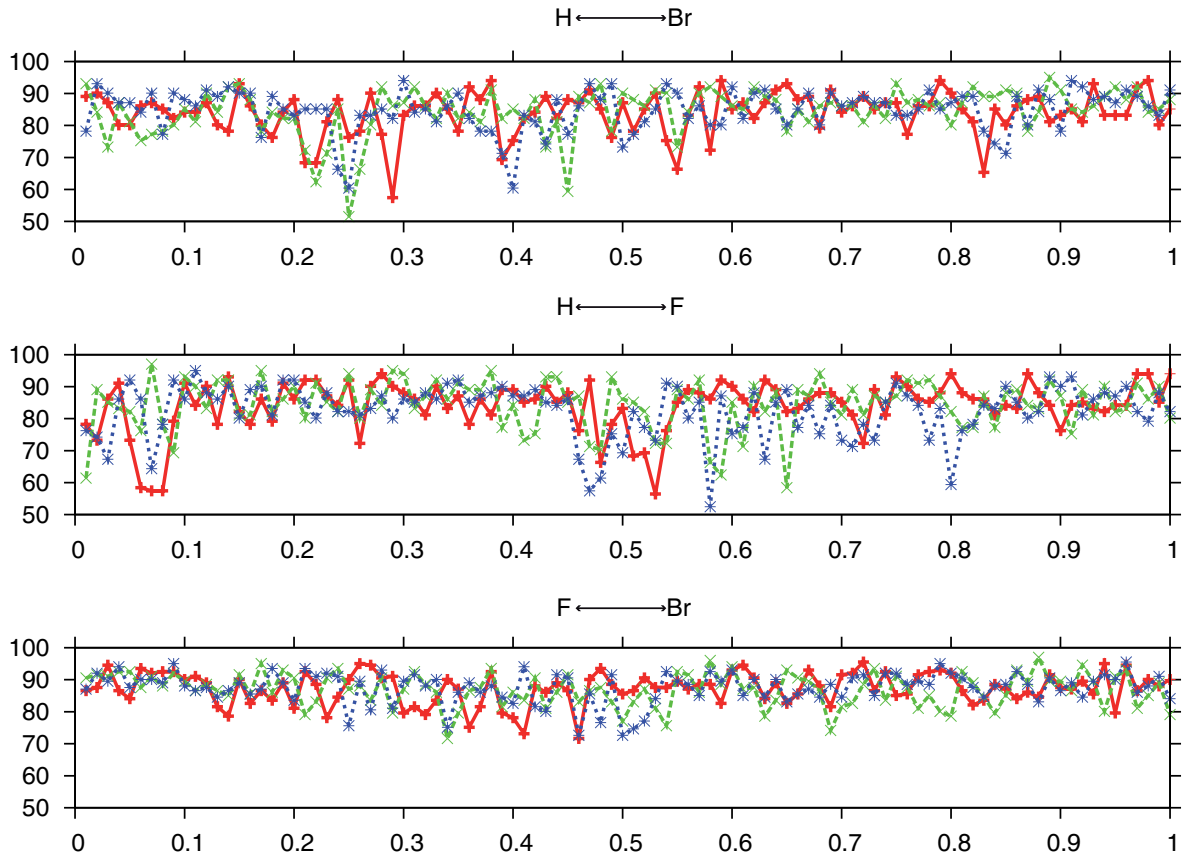


Figure 5.23.: Overlap of the probability distributions for the forward and backward transformation for all three different transformations in solvent. For each transformation three batches of data are presented. The overlap of the forward and backward transformations is calculated for all 100 windows and the respective three runs for each transformation. The x-axis shows the λ value for the different windows, the y-axis the overlap in %.

	H⟷Br	H⟷F	Br⟷F	Cycle
Run1	31.44	33.58	1.71	-0.43
Run2	31.79	33.68	1.66	-0.23
Run3	31.57	33.29	1.76	0.04
Average	**31.6 ± 0.14**	**33.52 ± 0.17**	**1.71 ± 0.04**	**-0.20 ± 0.19**

Table 5.7.: Results for the solvent transformations. For each transformation 3 runs were performed and the average was taken. Energies are given in kcal/mol

The results for the solvent transformations are very similar to the vacuum results. In all cases the mean absolute deviation between the different runs is about at maximum 0.4 kcal/mol (see Table 5.7). The fluorine shows a slightly higher free energy change (about 1-2 kcal/mol higher). Compared to the vacuum simulations the stability of the fluorinated inhibitor with regard to the brominated inhibitor is increased. The net relative solvation free energy for the transformation $Br \longleftrightarrow F$ is below 1.0 kcal/mol (-0.6 kcal/mol). This result is also confirmed by the two transformations starting from the hydrogen. The closed thermodynamic cycle again yields a very small hysteresis of about -0.2 $\pm$ 0.19 kcal/mol which is again very close to the ideal result of zero.
The results for the solvent transformations performed with CAST can be found in Table A.2 in the Appendix. Again the results are very close to the NAMD results. Absolute values for the $H \longleftrightarrow F$ and $F \longleftrightarrow Br$ transformations are nearly identical with NAMD results, while the $H \longleftrightarrow Br$ shows a small deviation of about 0.7 kcal/mol. This is also resembled in the value for the closed thermodynamic cycle which is calculated to -0.98 ± 0.31 kcal/mol which is still a good result. It has to be noted, that due to performance reasons only the forward transformations were calculated with CAST.

Protein transformations

The overlaps of the underlying probability distributions are very different for the three transformations (see Figure 5.24). For the $H \longleftrightarrow Br$ transformations (Figure 5.24 upper) the overlaps for runs 1 (red) and 2 (blue) are very poor, with drops down between 0 to 10% overlap. While transformation 1 shows a poor overlap especially over λ windows 0.0 to 0.6 the low overlap of the blue curve is between 0 and 0.4 with a reasonable overlap for the rest of the transformation. Only the third transformation (green) shows a good overlap over the whole transformation time. For the $H \longleftrightarrow F$ transformations the overlap is significantly better. All three simulations show a nice overlap until the midpoint of the transformation ($\lambda = 0.5$). Starting from the midpoint, the overlap gets worse for the blue and green curves (Figure 5.24 middle). The red curve however shows a very nice overlap over the whole simulation time. Unfortunately the third transformation, the $F \longleftrightarrow Br$ transformation also shows a very poor overlap. After the first calculation was finished and the overlap was plotted the other two transformations were canceled. Since this poor overlap of the distributions indicates a problem with the stratification scheme or the simulations itself, the trajectory of the $F \longleftrightarrow Br$ transformation was subjected to visual inspection. Since the stratification into 100 windows should result in extremely thorough sampling and very small free energy changes between neighboring windows, the more probable reason for the poor convergence is the simulation itself.

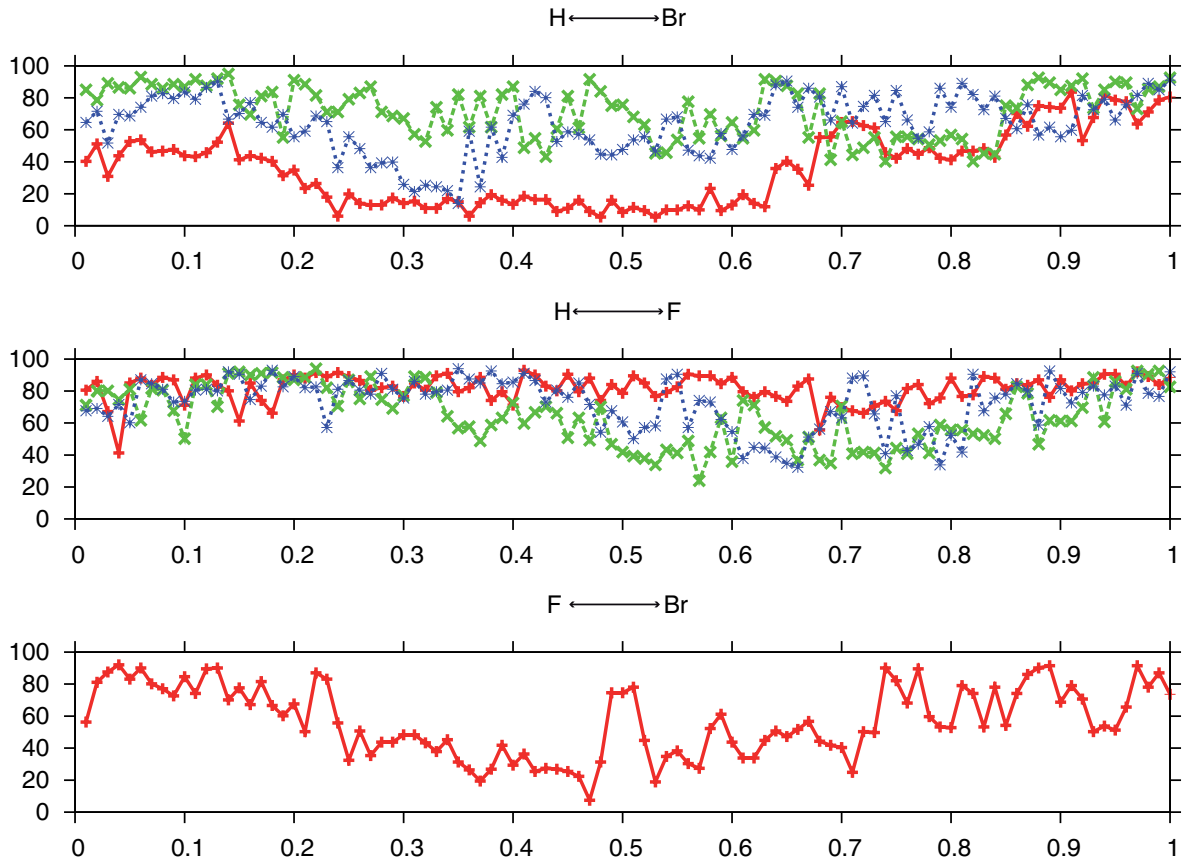
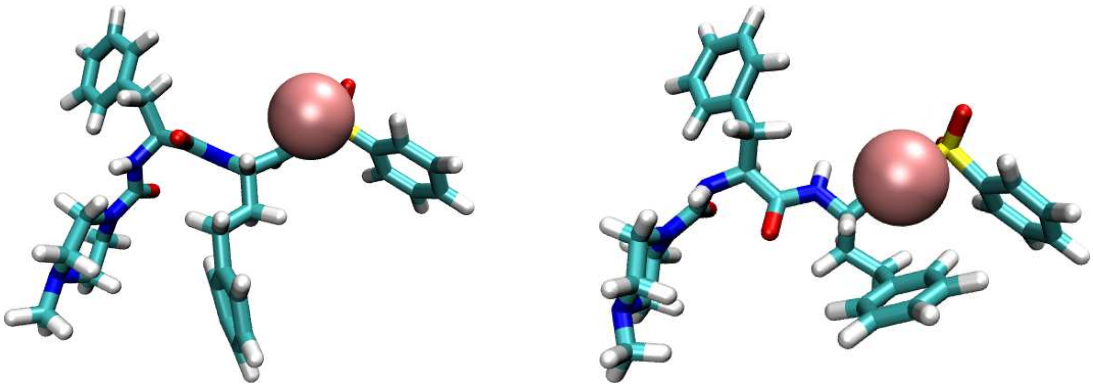


Figure 5.24.: Overlap of the probability distributions for the forward and backward transformation for all three different transformations in the protein environment. For $H \longleftrightarrow Br$ and $H \longleftrightarrow F$ three batches of data are presented. For the $Br \longleftrightarrow F$ only one data set is shown. The other transformations were cancelled at midpoint due to an already very poor overlap. The overlap between the forward and backward transformation is calculated for all 100 windows and all runs for the respective transformation. The x-axis show the λ value for the different windows, the y-axis the overlap in %.

As can bee seen from Figure 5.25, the left part of the inhibitor, namely the left phenyl unit, as well as the piperazin unit are moving only very little. In contrast the right part, especially the lower phenyl unit flips during the simulation in direction of the bromine atom at the end of the forward transition. To check when and how long the conformation stays during the transformation the distance between the bromine and the first carbon atom of the benzene ring is measured (see Figure 5.26). The flipping of the benzene ring occurs around snapshot 110 at which the bromine is about 75% present which is equal to a 75% completed forward transformation and amounts to roughly 15ns of simulation time. Since the equilibration was performed for a standard 5ns this conformation was not found during visual inspection. Since this conformation is only adopted at the very end of the forward transformation but all the way through the backward transformation

it is only very poorly sampled. The $F \longleftrightarrow Br$ transformation, has been restarted, this time with the inhibitor in the new conformation (Figure 5.25 right side) as the starting conformation. Again the calculation was performed a total of 3 times to get an average.



(a) Starting conformation **(b)** Final conformation

Figure 5.25.: Poses of the inhibitor at the start of the $F \longleftrightarrow Br$ transformation (a) and at the end of the transformation (b). The bromine atom is shown in the space filling representation.

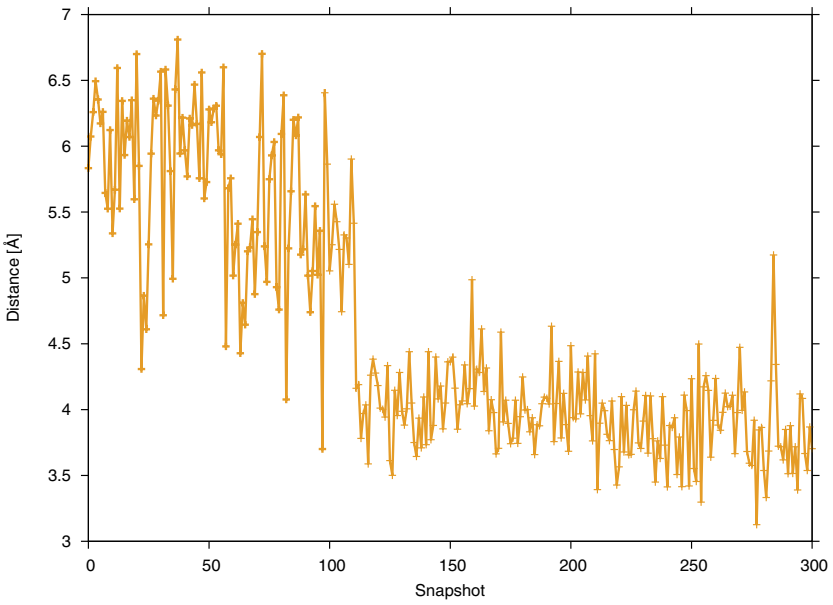


Figure 5.26.: Distance plot for the distance between the Bromine and the first carbon of the middle benzene ring for the $F \longleftrightarrow Br$ transformation. The forward transformation ($F \rightarrow Br$) is completed at snapshot 150. Starting from this point the bromine is transformed back to the fluorine.

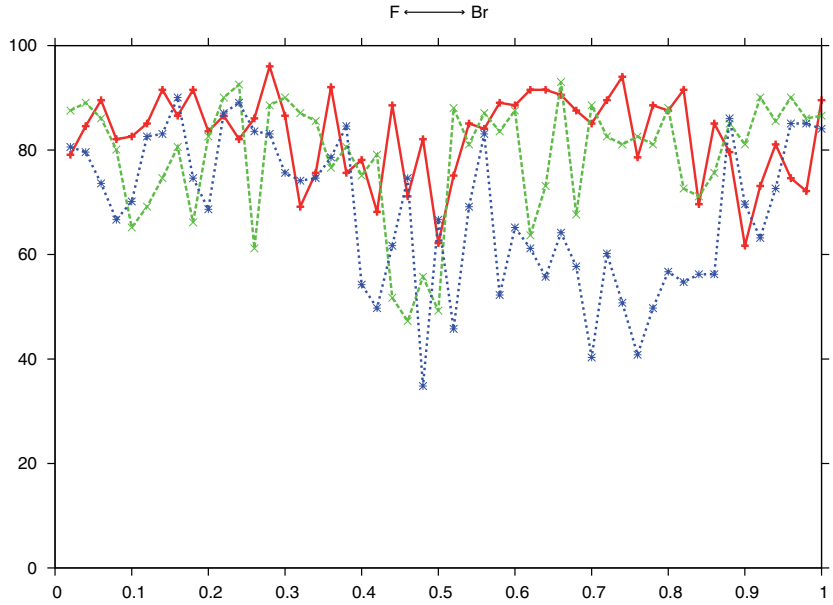


Figure 5.27.: Overlap of the probability distributions for the forward and backward transformation for $Br \longleftrightarrow F$ starting with the new conformation (see Figure 5.25 right side). Three batches of data are presented. The overlap between the forward and backward transformation is calculated for all 100 windows and all three runs. The x-axis show the λ value for the different windows, the y-axis the overlap in %.

Comparing Figure 5.27 with the former results in Figure 5.24 shows a dramatic increase in the overlap of the probability distributions and therefore convergence. Looking at the overall performance of the protein simulations, deviations between the single runs are bigger compared to vacuum or solvent (MAD between 1-2 kcal/mol). Overall the fluorine substituted inhibitor shows a slightly higher free energy change of about 2.5 kcal/mol than the bromine substituted inhibitor.

Protein transformations

	H⟷Br	H⟷F	Br⟷F	Cycle
Run1	31.49	36.01	2.39	-2.13
Run2	34.52	34.53	2.00	1.99
Run3	33.78	33.92	3.15	3.01
Average	**33.26 ± 1.29**	**34.82 ± 0.88**	**2.51 ± 0.48**	**0.95 ± 2.22**

Table 5.8.: Results for the protein transformations. For each transformation 3 runs were performed and the average was taken. Energies are given in kcal/mol.

Free energy of binding

The relative binding free energies are depicted in Table 5.9. The first two rows of Table 5.9 show the results for the free energy change of the respective computations averaged over the respective three runs, regardless of the poor overlap distributions. The third row shows the relative binding free energy calculated from row 1 and 2. The row with ΔG^2_{alch} (best) shows the relative free energy changes for the respective transformations in the protein environment.

Relative binding free energies

Energy	H$\longleftrightarrow$Br	H$\longleftrightarrow$F	Br$\longleftrightarrow$F	Cycle
ΔG^1_{alch} (solv)	31.51	33.51	1.71	-0.29
ΔG^2_{alch} (prot)	33.26	34.82	2.51	0.64
$\Delta\Delta G$	**1.75**	**1.30**	**0.80**	**1.25**
ΔG^2_{alch} (best)	33.78	36.01	2.51	0.28
$\Delta\Delta G$	**2.27**	**2.5**	**0.80**	**0.57**

Table 5.9.: Results for the relative binding free energies for the different inhibitors based on the K11777. Rows 1 to 2 show the relative free energies in solution and in the protein, calculated form the average over all runs. The relative binding free energy in row three is calculated from these values. Row 4 shows the relative free energies in the protein environment based on the run with the best overlap of probability distributions. The relative free energy of binding in row 5 is calculated from the difference of row 4 and row 1. Energies are given in kcal/mol.

In this case however, the values correspond to the transformations with the best overlap of the probability distributions. For the $H \longleftrightarrow Br$ transformation (Figure 5.24 upper plot) the green curve was used, for the $H \longleftrightarrow F$ transformation (Figure 5.24 middle plot) the red curve was used and for the $Br \longleftrightarrow F$ transformation (Figure 5.27) also the red curve was used. The relative free energy of binding was then calculated from these values and the average of the relative free energies obtained in solution (Table 5.9 first row).

The overall differences between the inhibitors are very small to non existent and the difference in relative binding free energy is smaller than the deviations between some of the protein runs. According to the calculations from the averages, the relative binding free energy for the $H \longleftrightarrow Br$ transformation amounts to about 1.8 kcal/mol whereas the $H \longleftrightarrow F$ transformation is slightly more favorable with 1.3 kcal/mol. Comparing these results with the better converged direct transformation $Br \longleftrightarrow F$ shows that the net difference in the binding affinities for $Br \longleftrightarrow F$ of 0.8 kcal/mol in favor of bromine

is not recovered. According to the simulations starting from the hydrogen, the inhibitor carrying the fluorine is the more stable inhibitor. Since the direct $F \longleftrightarrow Br$ simulations show a far better convergence, the averaged results for the simulations starting from the hydrogen have to be taken with caution.

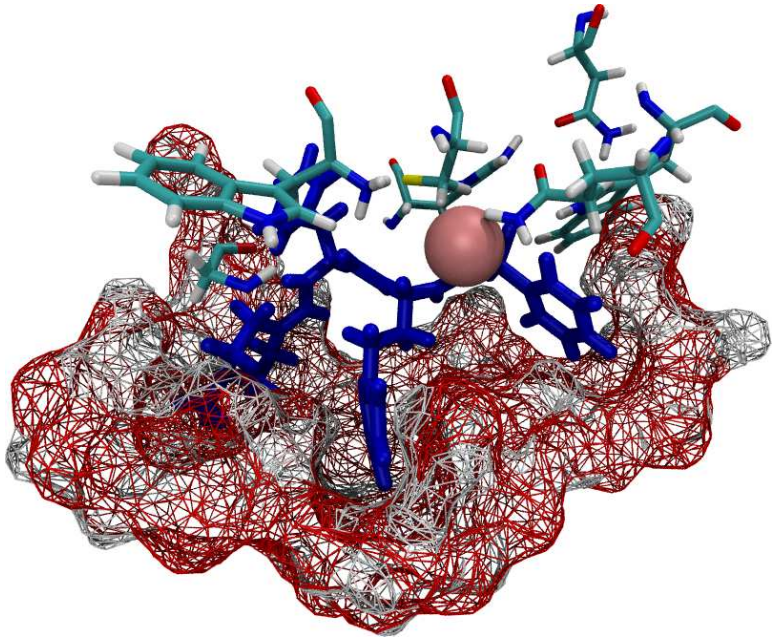


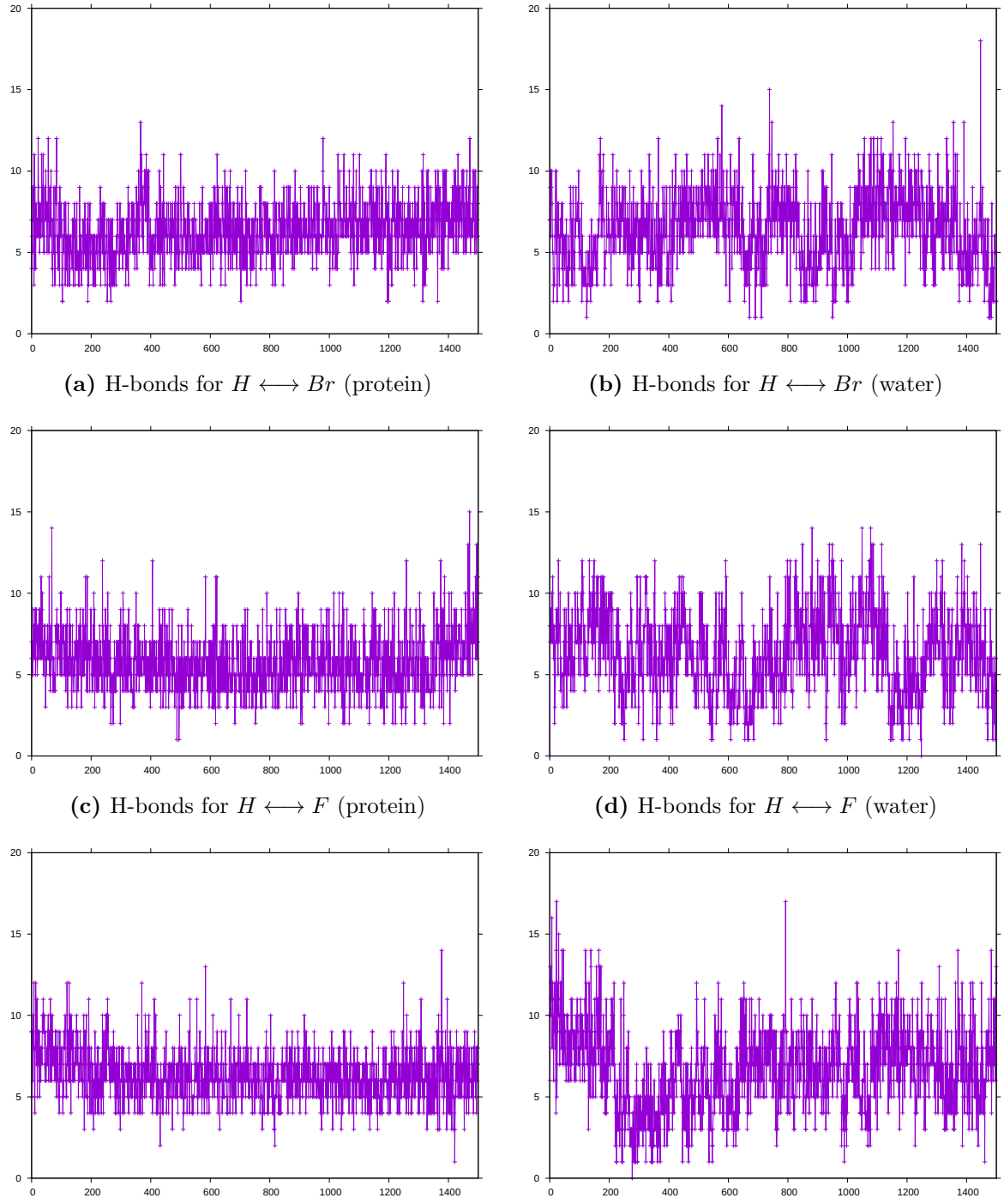
Figure 5.28.: Depiction of the inhibitor (blue) substituted with a bromine (space filling) and the first shell of neighboring amino acids and water (red wireframe).

If only the results for the transformations with the best overlap in probability distribution are used, the trend is reveresed and also reflected when starting the calculations from the original K11777 inhibitor. In this case the $H \longleftrightarrow F$ transformation is with 2.5 kcal/mol slightly less favorable than the $H \longleftrightarrow Br$ transformation with 2.3 kcal/mol. This trend is also recovered in the direct $Br \longleftrightarrow F$ calculation which amounts to 0.7 kcal/mol. The closed cycle then yields a net free energy change of 0.6 kcal/mol. In order to understand the similarities in the energetics, a closer look at several possible effects for this behavior was taken. In Figure 5.28, the orientation of the inhibitor in the binding pocket is depicted. Since the inhibitor is partly exposed to water, the hydrogen bonding pattern may play a role in the binding affinities. Furthermore, a closer look at the electrostatic potentials and possible steric interactions have been taken.

Hydrogen bonds

First, the hydrogen bonding pattern for each of the different inhibitors was inspected. Hydrogen bonds were examined for the calculations in solvent and for the simulations

in the protein environment. A hydrogen bond between the inhibitor and water or the inhibitor and the protein was assigned if the distance for the hydrogen bond was below 3Å and the hydrogen bond angle cutoff was less than 90°. For the solvent simulations 3 batches of data were used, each consisting of 1500 snapshots. The same amount of snapshots was used for the results of the protein simulations. Hydrogen bonds were examined for the $H \longleftrightarrow Br$, $H \longleftrightarrow F$ and $F \longleftrightarrow Br$ transformations. A representative plot, featuring the number of hydrogen bonds formed between inhibitor and protein or inhibitor and water, for the three transformations is depicted in Figure 5.29. All plots cover the complete bidirectional transformation of the inhibitors in one another. In the aqueous-phase during the $H \longleftrightarrow Br$, the inhibitor exerts 6.38 ± 2.26 hydrogen bonds over the simulation time. For the same simulation performed in the protein the average number of hydrogen bonds amounts to 6.34 ± 1.76. Moving the inhibitor from the aqueous phase to the protein would result in the loss of approximately 0.04 hydrogen bonds which is in the range of the statistical uncertainty. For the $H \longleftrightarrow F$ transformation the inhibitor shows 6.28 ± 2.47 hydrogen bonds in the water and 5.78 ± 1.81 hydrogen bonds in the protein. The loss of hydrogen bonds is abut 0.5 which also is in the range of the statistical uncertainty. The trend observed during the simulations starting from the K11777 is reproduced during the $F \longleftrightarrow Br$ transformation. An average of 6.64 ± 2.54 hydrogen bonds in water, and an average number of 6.35 ± 1.86 hydrogen bonds in the protein lead to a loss of approximately 0.3 hydrogen bonds going from the aqueous phase to the protein. In sum, the total number and loss of hydrogen bonds is equal for all transformations and independent of the substitution pattern. This also supports the similarities found in the energetics.



(a) H-bonds for $H \longleftrightarrow Br$ (protein)

(b) H-bonds for $H \longleftrightarrow Br$ (water)

(c) H-bonds for $H \longleftrightarrow F$ (protein)

(d) H-bonds for $H \longleftrightarrow F$ (water)

(e) H-bonds for $Br \longleftrightarrow F$ (protein)

(f) H-bonds for $Br \longleftrightarrow F$ (water)

Figure 5.29.: Variation of the number of hydrogen bonds of X =H, F, Br along the respective bidirectional transformation. The left column gives the protein environment while the right column gives the solvent. The 1500 snapshots cover the whole transformation from $\lambda = 0$ to $\lambda = 1$ back to $\lambda = 0$. For each transformation one representative run out of three (solvent and protein) is shown.

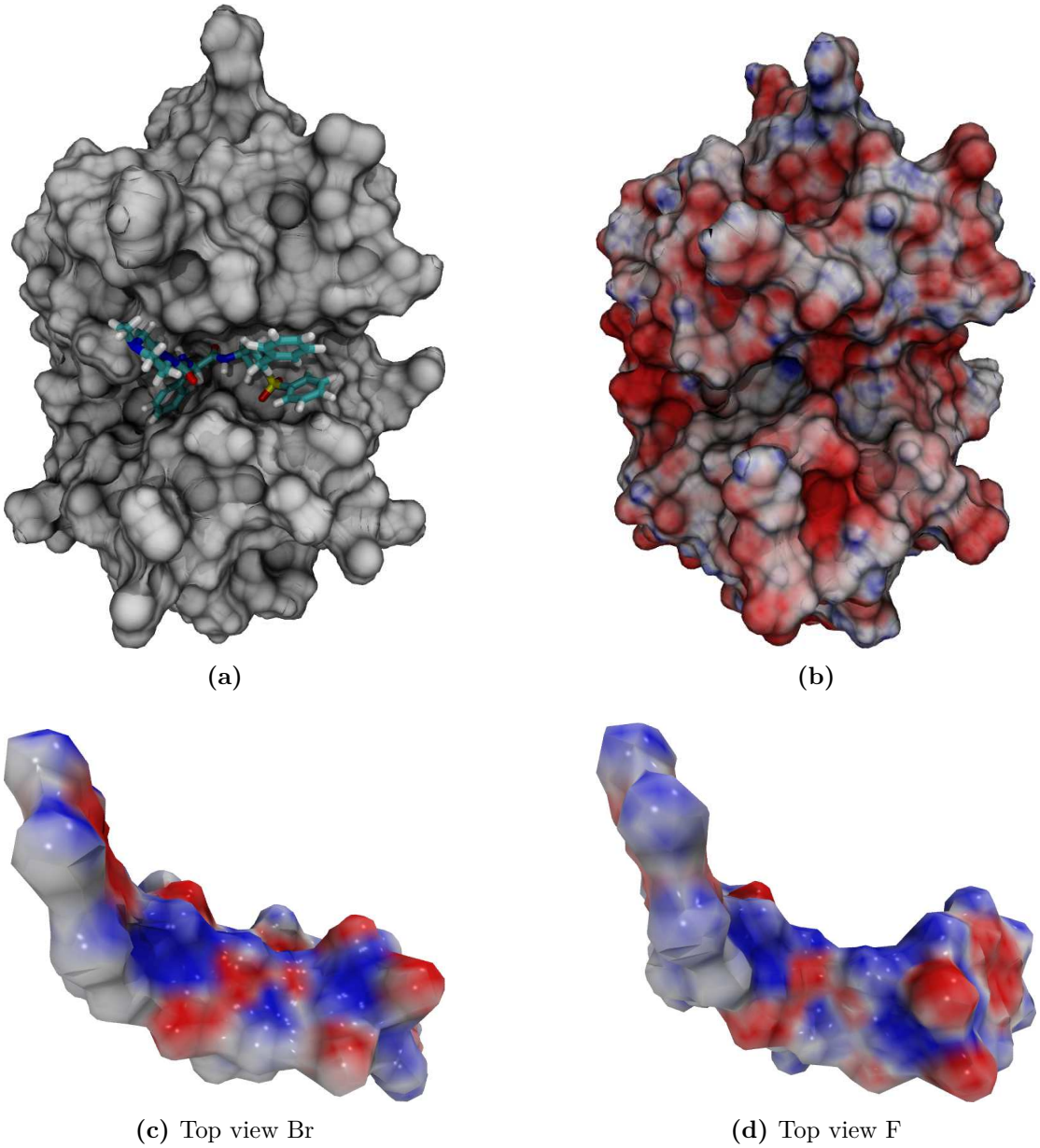
Electrostatics

Since the catalytic dyad in the protein is charged, the electrostatic potential may have an influence on the binding affinity of the inhibitors. Table 5.10 shows the dipole moments of the inhibitors. The dipole moments were calculated with Turbomole 7.0 [305] using the B3-LYP functional and a cc-pVTZ [306,307,307,308] basis set. For the treatment of the implicit solvation, the COSMO model was applied. For the calculation of the dipole moments in water a value of $\varepsilon = 78.355$, for the calculations in the protein environment a value of $\varepsilon = 5$ was used. Assuming purely continuum electrostatics, the free energy of solvation should become more negative the higher the dipole moment of the molecule is [309].

	Dipole moment [Debye]	
Inhibitor	Water environment	Protein environment
Inh-H	10.4	6.72
Inh-F	9.81	5.57
Inh-Br	9.54	5.67

Table 5.10.: Dipole moments of the K11777, fluorinated and brominated inhibitors in solvent and in the protein environment. Dipole moments are given in Debye.

This is in nice agreement with the calculations. The free energy of hydration is 4.69 kcal/mol higher for the bromine substituted inhibitor compared to the K11777. For the fluorine substituted inhibitor the free energy of hydration 4.44 kcal/mol higher. This implies that the inhibitor with the fluorine substituent is 0.25 kcal/mol more stable in water than the bromine substituted. This result is also recovered with the direct transformation which shows a difference in relative hydration free energy of about -0.57 kcal/mol in favor of the fluorinated inhibitor. In accordance, the influence of the surrounding medium on the free energy increases the higher the dielectric constant of the medium is. The dielectric constant of water is about 80 [310], whereas a protein has a dielectric constant of 2 to 30 [311, 312]. As shown in Figure 5.28, the bottom part of the inhibitor is exposed to water. If one changes the inhibitor and with it its dipole moment, a response of the solvent to this change can be expected. Compared to the simulations in aqueous solution, this should however be smaller in the protein environment. If all other effects are assumed to be equal in strength, the most polar inhibitor is supposed to be the weakest binder, and vice versa, the least polar one should be strongest. Figure 5.30 shows the electrostatic potential of the protein as well as the inhibitors.



(a) **(b)**

(c) Top view Br **(d)** Top view F

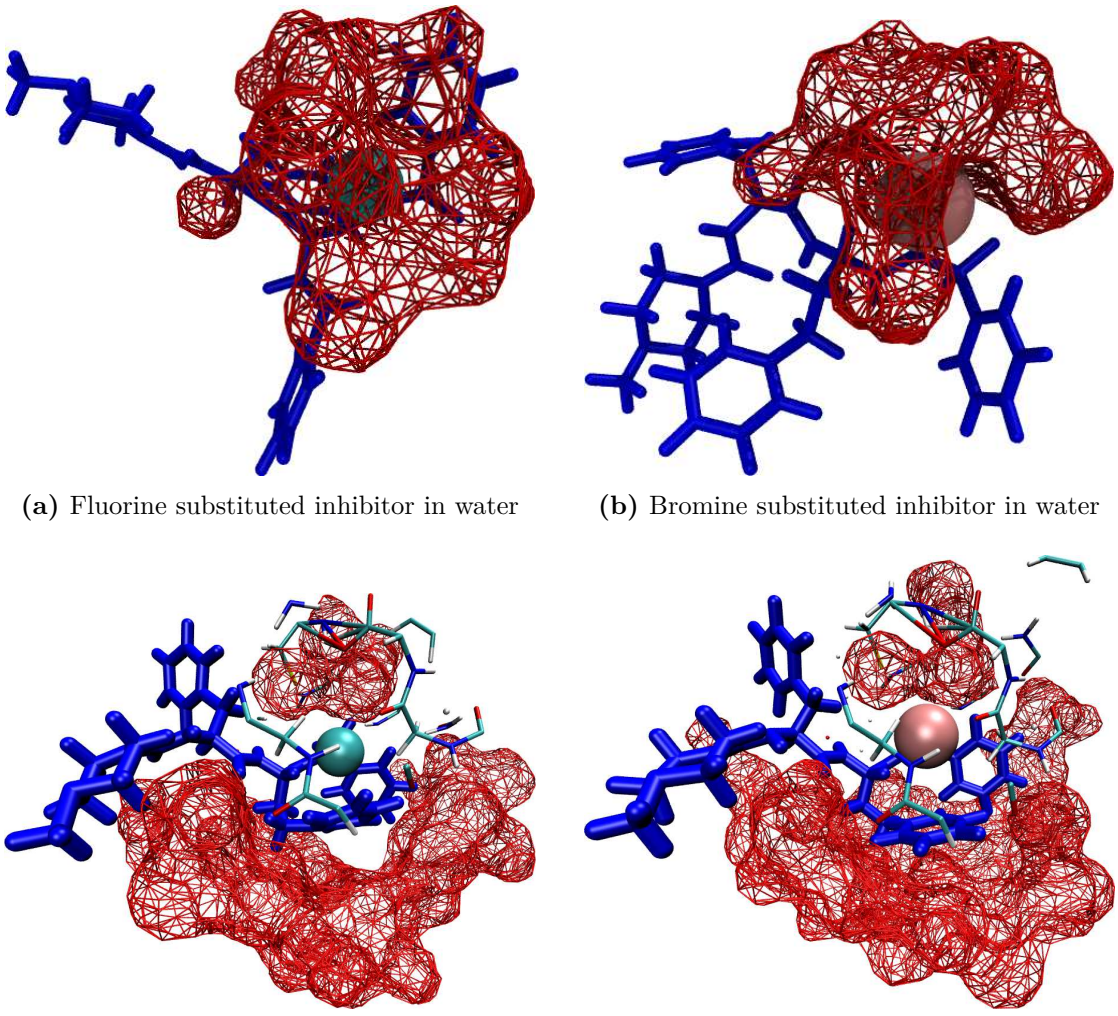
Figure 5.30.: Depiction of the electrostatic surface of the protein and the inhibitors. Figure (a) shows the inhibitor embedded in the active site. Figure (b) shows the electrostatic potential of the protein. Figures (c) and (d) show the electrostatic potential maps for the brominated, respective fluorinated inhibitors. Positive values are shown in blue, negative in red.

The electrostatic potentials were calculated using the Adaptive Poisson-Boltzmann Solver [313] (APBS) with the CHARMM force field parameters to be in accordance with the other calculations. In Figure 5.30 (c) and (d) the inhibitors with bromine (c) and fluorine (d) are shown from the top. The top side in this case means the side which points to the protein when sitting in the binding pocket. Figure 5.30 (a) shows the inhibitor sitting in the binding pocket, and Figure 5.30 (b) shows the protein in

the same pose with its electrostatic potential map. The poses of the inhibitors were chosen randomly from the last window of the forward FEP simulations where only the target system is present. When looking at the electrostatic potential map of the protein one notices that the binding pocket is predominantly neutral with a negatively charged part, which is due to the negative charge on the cystein residue, which is responsible for the attack during the formation of a covalent bond with the inhibitor. Looking at the inhibitor maps, both show a nice complementary agreement in electrostatic potential. The potential maps also allow a rough estimate of the polarity of the molecules. If one now judges the inhibitor strength by the dipole moments the free energy of binding should be Inhib-Br < Inhib-F which is also the result observed in experiment. Nevertheless the differences between the three inhibitors are extremely small.

Sterics

The third effect that was inspected results from possible sterical interactions between the inhibitors and the surrounding medium. Since fluorine and hydrogen have almost the same van-der-Waals radius, the biggest difference is expected between H (or F) and Br. The poses of the inhibitors in the active site and in the water are depicted in Figure 5.31. Looking at Figure 5.31 (a) and (b) it becomes clear that the movement and the conformational space of the inhibitor is not at all, or only very weakly, restricted in the water environment. The conformations exerted cover almost the whole range allowed by the flexibility of the respective bonds and angles regardless of the substitution pattern. The picture clearly changes in the protein (Figure 5.31 (c) and (d)). Here both inhibitors exert almost the same conformation during the transformation. No large movements can be detected for the inhibitor, only minor deviations in the orientation of some sidechains of the protein are visible. However, the most distinct features are first, the shielding of the bromine, respective fluorine from the bottom by the phenyl ring and second the access of water from the top. The shielding by the phenyl ring can be found for both, the fluorine and the bromine substituents over the whole simulation time. When the fluorine is present, the phenyl ring can flip a little bit more in direction of the substituent since the van-der-Waals radius of the fluorine is much smaller. Nevertheless, the effect is the same for both halogen atoms. The atom gets shielded from below. This only allows one possibility for water to access the atoms, namely from the top. Furthermore, the protein surroundings of both inhibitors are nearly identical as is the space to the amino acids and the water molecules. The much larger van-der-Waals radius of the bromine atom has only very little to no effect since the distance between the atom and the surrounding does not change when going from fluorine to bromine or vice versa.



(a) Fluorine substituted inhibitor in water

(b) Bromine substituted inhibitor in water

(c) Fluorine substituted inhibitor in protein environment

(d) Bromine substituted inhibitor in protein environment

Figure 5.31.: Poses of the inhibitors in the active site. The pictures were taken randomly from the direct $F \longleftrightarrow Br$ transformation. The inhibitor is shown in blue, with the respective fluorine (left) or bromine (right) in the space filling representation. The surrounding water is depicted as a red wireframe surface.

Conclusion The biggest change in energetics is expected when going from vacuum to the solvent. The differences between the new inhibitors and the K11777 are in the range of about 4-5 kcal/mol and the differences between the new inhibitors are less than 1 kcal/mol which implies equal interactions for hydrogen, fluorine and bromine with the surrounding water molecules when the molecules are solvated. The similarity of the interactions is supported by the nearly identical number of hydrogen bonds in the solvent. When moving the inhibitors from the solvent to the protein environment the change is expected to be smaller in total which is also supported by the calculations. The possible loss of hydrogen bonds is equal for all three inhibitors. Furthermore, the

electrostatic potential map is nearly identical for the fluorine substituted and bromine substituted inhibitors. Both show a nice agreement with the electrostatic potential map of the protein. The relative binding free energy between the K11777 and the new inhibitors has been calculated to 2.27 kcal/mol (Br) and 2.5 kcal/mol (F) respectively and shows the assumed smaller change for the solvent → protein move compared to the vacuum → solvent move. The similarities for both new inhibitors are further supported by the orientation of the inhibitors in the active site. Looking at Figure 5.31, both inhibitors and especially the differing functional groups are buried in the active site in the same manner. This also results in a rather similar relative binding free energy with a difference of approximately 0.6 kcal/mol. In sum, the similarities in the energetics of the three inhibitors can be attributed to the similarities in hydrogen bonding pattern, electrostatics, orientation and possible steric interactions of the compounds in the active site. Concluding, the performed calculations can not explain the experimental results which show the fluorine substituted inhibitor leaving the active site much faster than the bromine substituted inhibitor.

6. Summary

The aim of the present work is the development and implementation of new simulation possibilities for the CAST program package. Development included, among other things, the partial parallelization of the already existing force fields, extension of the treatment of electrostatic interactions and implementation of molecular dynamics and free energy algorithms.
The most time consuming part of force field calculations is the evaluation of the non-bonded interactions. The calculation of these interactions has been parallelized and it could be shown to yield a significant speed up for multi-core calculations compared to the serial execution on only one CPU. For both, simple energy/gradient as well as molecular dynamics simulations the computational time could be significantly reduced. To further increase the performance of calculations employing a cutoff radius, a linked-cell algorithm was implemented which is able to build up the non-bonded interaction list up to 7 times faster than the original algorithm.
To provide access to dynamic properties based on the natural time evolution of a system, a molecular dynamics code has been implemented. The MD implementation features two integration schemes for the equations of motion which are able to generate stable trajectories. The basic MD algorithm as described in Section 1.2 leads to the sampling in the microcanonical (NVE) ensemble. The practical use of NVE simulations is limited though because it does not correspond to any experimentally realistic situation. More realistic simulation conditions are found in the isothermal (NVT) and isothermal-isobaric (NPT) ensembles. To generate those ensembles, temperature and pressure control has been implemented. The temperature can be controlled in two ways: by direct velocity scaling and by a Nose-Hoover thermostat which produces a real canonical ensemble. The pressure coupling is realized by implementation of a Berendsen barostat. The pressure coupling can be used for isotropic or anisotropic box dimensions with the restriction that the angles of the box need to be 90°. A crucial simulation parameter in MD simulations is the length of the timestep. The timestep is usually in the rang of 1fs. Increasing the timestep beyond 1fs can lead to unstable trajectories since the fastest motion in the system, usually the H-X stretch vibration can not be sampled anymore. A way to allow for bigger timesteps is the use of a constraint algorithm which constrains

the H-X bonds to the equilibrium distance. For this the RATTLE algorithm has been implemented in the CAST program. The velocity Verlet algorithm in combination with the RATTLE algorithm has been shown to yield stable trajectories for an arbitrary length of simulation time. In a first application the MD implementation is used in conjunction with the MOPAC interface for the investigation of PBI sidechains and their rigidity. The theoretical investigations show a nice agreement with experimentally obtained results. Based on the MD techniques two algorithms for the determination of free energy differences have been implemented. The umbrella sampling algorithm can be used to determine the free energy change along a reaction coordinate based on distances or dihedral angles. The implementation was tested on the stretching of a deca-L-alanine and the rotation barrier of butane in vacuum. The results are in nearly perfect agreement with literature values. For the FEP implementation calculations were performed for a zero-sum transformation of ethane in explicit solvent, the charging of a sodium ion in explicit solvent and the transformations of a tripeptide in explicit solvent. All results are in agreement with benchmark calculations of the NAMD program as well as literature values. The FEP formalism was then applied to determine the relative binding free energies between two inhibitors in an inhibitor-protein complex.
Next to force fields, *ab-initio* methods can be used for simulations and global optimizations. Since the performance of such methods is usually significantly poorer than force field applications, the use for global optimizations is limited. Nevertheless significant progress has been made by porting these codes to GPUs. In order to make use of these developments a MPI interface has been implemented into CAST for communication with the DFT code TeraChem. The CAST/TeraChem combination has been tested on the H_2O_{10} cluster as well as the polypeptide met-Enkephalin. The pure *ab-initio* calculations showed a superior behavior compared to the standard procedure where the force field results are usually refined using quantum chemical methods.

7. Zusammenfassung

Das Ziel der hier vorliegenden Arbeit ist die Entwicklung und Implementierung neuer Simulationsalgorithmen in das CAST Programmpaket. Neben der teilweisen Parallelisierung der bereits impelentierten Kraftfelder wurde das Programm um einen Molekulardynamikcode erweitert. Aufbauend auf diesem Code wurden Algorithmen zur Berechnung der freien Energie entlang einer Reaktionskooridnate, sowie eine Erweiter-ung der Behandlung elektrostatischer Wechselwirkungen auf Basis einer Ewald Summation implementiert.
Der zeitaufwändigste Teil einer Kraftfeldrechnung stellt die Evaluierung der nichtbindenden Wechselwirkungen dar. Die Berechnung dieser Wechselwirkungen wurde für die Nutzung von Mehrkernprozessoren optimiert und parallelisiert. Die Parallelisierung zeigte eine signifikante Reduktion der benötigten Rechenzeit auf mehreren Rechenkernen im Vergleich zur seriellen Berechnung auf nur einem Rechenkern für einfache Energie- bzw. Gradientenrechnungen sowie für Molekulardynamikrechnungen. Um Rechnungen, die einen cutoff Radius benutzen, weiter zu beschleunigen, wurde der Aufbau der Verlet-Liste modifiziert. Statt des Standardalgorithmus, der eine Doppelschleife über alle Atome verwendet, wurde ein linked-cell Algorithmus implementiert. Der Aufbau der Verlet-Liste konnte damit um den Faktor 7 beschleunigt werden.
Der Molekulardynamikcode enthält mehrere Algorithmen zur Berechnung von Systemen in verschiedenen Ensembles. Zur numerischen Integration der Bewegungsgleichungen wurden der Velocity-Verlet sowie eine modifizierte Version von Beemans Algorithmus implementiert. Da der minimale Code, wie er in Kapitel 1.2 beschrieben wird, ein mikrokanonisches Ensemble produziert, und dieses keiner realistischen experimentellen Situation entspricht, wurden Methoden zur Berechnung und Aufrechterhaltung von Temperatur und Druck implementiert. Die Temperatur kann mittels zweier verschiedener Möglichkeiten kontrolliert werden. Die erste Möglichkeit ist die direkte Skalierung der Geschwindigkeiten der Partikel, die zweite Möglichkeit besteht in der Nutzung eines Nòse-Hoover Thermostaten, der ein echtes kanonisches Ensemble generiert. Für die Kontrolle des Drucks wurde ein Berendsen Barostat implementiert. Da die Kontrolle des Drucks die Nutzung von periodischen Randbedingungen voraussetzt, ist die Form der Simulationszelle wichtig. CAST unterstützt aktuell isotrope und anisotrope Simu-

lationszellen, mit der Einschränkung, dass alle Winkel 90°betragen.
Ein kritischer Punkt bei einer MD Simulation ist die Länge des Zeitschritts, der in der Regel bei 1fs liegt. Sollen größere Zeitschritte verwendet werden, müssen die schnellsten Bewegungnen im System eingeschränkt werden. Dies sind im Normalfall die H-X Streckschwingungen. Zur Einschränkung dieser wurde der RATTLE Algorithmus implementiert der die H-X Bindung mit Hilfe von Lagrange-Multiplikatoren auf den Gleichgewichtsabstand fixiert. Als erste Anwendung des MD Codes wurde in Kombination mit dem MOPAC Programm die Rigidität und Flexibilität von PBI Seitenketten erfolgreich untersucht.
Basierend auf dem MD Code wurden zwei Möglichkeiten zur Bestimmung der freien Energie eingebaut, Umbrella Sampling und Free Energy Perturbation. Umbrella Sampling erlaubt die Bestimmung der freien Energie entlang einer Reaktionskoordinate, hier Abstände oder Diederwinkel. Der Algorithmus wurde erfolgreich an zwei Literaturbeispielen, der Streckung von Deca-L-Alanin sowie der Rotation von Butan um den zentralen Diederwinkel getetstet. Die FEP Implementierung wurde an drei Beispielen getestet, einer Nullsummen-Transformation von Ethan in Ethan in explizitem Solvent, dem Lösen eines Natriumions in Wasser und der Transformation von Tyrosin in Alanin in einem Tripeptid. Die Ergebnisse dieser Testrechnungen stimmen hervorragend mit Vergleichsrechnungen mit NAMD sowie Literaturwerten überein. Die FEP Methode wurde schließlich benutzt um die relative freie Bindungsenergie zweier Inhibitoren in einem Inhibitor-Protein-Komplex zu bestimmen.
Neben Kraftfeldern können auch *ab-initio* Methoden für Simulationen benutzt werden. Da die Rechenzeit dieser Methoden um ein vielfaches höher ausfällt als die für Kraftfelder, ist die Benutzung für die globale Optimierung jedoch limitiert. In den letzten Jahren wurde im Bereich der Leistungsfähigkeit dieser Methoden jedoch große Fortschritte erzielt, indem diese Methoden auf Grafikkarten portiert wurden. Um diese Entwicklung nutzbar zu machen wurde eine MPI-Schnittstelle in CAST implementiert, die mit dem DFT Programm TeraChem kommuniziert. Die Kombination aus beiden Programmen, sowie die Funktionsfähigkeit der Schnittstelle, wurde an H_2O_{10} Clustern sowie dem Polypeptid Met-Enkephalin getestet. Die reinen *ab-initio* Rechnungen zeigten ein besseres Verhalten gegenüber dem Normalen Protokoll, welches Kraftfeldrechungen mit nachfolgender Optimierung mit qunatenchemischen Methoden vorsieht.

Bibliography

[1] Adcock, A. S.; McCammon, J. A. *Chem. Rev.* **2006**, *106*, 1589–1615.

[2] Senn, H. M.; Thiel, W. *Angew. Chem. Int. Ed.* **2009**, *48*(7), 1198–1229.

[3] Hansen, N.; van Gunsteren, W. F. *J. Chem. Theory Comput.* **2014**, *10*(7), 2632–2647.

[4] Frenkel, D.; Smit, B. *Understanding Molecular Simulation;* Academic Press, Inc.: Orlando, FL, USA, 2nd ed., **2001**.

[5] Allen, M. P.; Tildesley, D. J. *Computer simulation of liquids;* Oxford University Press, **1989**.

[6] Becker, O. M.; MacKerell Jr., A. D.; Roux, B.; Watanabe, M. *Computational Biochemistry and Biophysics;* Taylor & Francis, **2001**.

[7] Parr, R. G.; Craig, D. P.; Ross, I. G. *J. Chem. Phys.* **1950**, *18*(12), 1561–1563.

[8] Chen, T. C. *J. Chem. Phys.* **1955**, *23*(11), 2200–2201.

[9] Roothaan, C. C. J. *J. Chem. Phys.* **1958**, *28*(5), 982–983.

[10] Allen, L. C.; Karo, A. M. *Rev. Mod. Phys.* **1960**, *32*, 275–285.

[11] Parr, R. G. *Int. J. Quantum. Chem.* **1990**, *37*(4), 327–347.

[12] Dewar, M. J. S.; Thiel, W. *J. Am. Chem. Soc.* **1977**, *99*(15), 4899–4907.

[13] Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. *J. Am. Chem. Soc.* **1985**, *107*(13), 3902–3909.

[14] Stewart, J. J. P. *J. Comput. Chem.* **1989**, *10*(2), 209–220.

[15] Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M. *J. Am. Chem. Soc.* **1992**, *114*(25), 10024–10035.

[16] Westheimer, F. H.; Mayer, J. E. *J. Chem. Phys.* **1946**, *14*, 733–738.

[17] Westheimer, F. H.; Mayer, J. E. *J. Chem. Phys.* **1947**, *15*, 252–260.

[18] Rieger, M.; Westheimer, F. H. *J. Am. Chem. Soc.* **1950**, *72*, 19–28.

[19] Allinger, N. L.; Norman, L. *J. Am. Chem. Soc.* **1977**, *99*(25), 8127–8134.

[20] Allinger, N. L.; Flanagan, H. L. *J. Comput. Chem.* **1983**, *4*(3), 399–403.

[21] Allinger, N. L.; Pathiaseril, A. *J. Comput. Chem.* **1987**, *8*(8), 1225–1231.

[22] Allinger, N. L.; Yuh, Y. H.; Lii, J. H. *J. Am. Chem. Soc.* **1989**, *111*(23), 8551–8566.

[23] Lii, J. H.; Allinger, N. L. *J. Am. Chem. Soc.* **1989**, *111*(23), 8566–8575.

[24] Lii, J. H.; Allinger, N. L. *J. Am. Chem. Soc.* **1989**, *111*(23), 8576–8582.

[25] Gibson, K. D.; Scheraga, H. A. *Proc. Natl. Acad. Sci.* **1967**, *58*, 420–427.

[26] Gibson, K. D.; Scheraga, H. A. *Proc. Natl. Acad. Sci.* **1967**, *58*, 1317–1323.

[27] Scott, R. A.; Vanderko.G.; Tuttle, R. W.; Shames, P. M.; Scheraga, H. A. *Proc. Natl. Acad. Sci.* **1967**, *58*, 2204–2211.

[28] Lifson, S.; Warshel. *J. Chem. Phys.* **1969**, *49*, 5116–5129.

[29] Warshel, A.; Lifson, S. *Chem. Phys.* **1969**, *4*, 255–256.

[30] Warshel, A.; Lifson, S. *J. Chem. Phys.* **1970**, *53*, 582–594.

[31] Warshel, A.; Levit, M.; Lifson, S. *J. Chem. Phys.* **1970**, *33*, 84–99.

[32] Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. *J. Chem. Phys.* **1953**, *21*(6), 1087–1092.

[33] Hastings, W. K. *Biometrika* **1970**, *57*(1), 97–109.

[34] Andrieu, C.; De Freitas, N.; Doucet, N. A.; Jordan, M. I. *Mach. Learn.* **2003**, *50*, 5–43.

[35] Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. *Science* **1983**, *220*(4598), 671–680.

[36] Barker, A. A. *Austr. J. Phys.* **1965**, *18*, 119–133.

[37] Peskun, P. H. *Biometrika* **1973**, *60*(1), 607–612.

[38] Smith, R. L. *Oper. Res.* **1984**, *32*(6), 1296–1308.

[39] Chen, M.-H.; Schmeiser, B. W. *Oper. Res. Lett.* **1996**, *19*(4), 161 – 169.

[40] Nayeem, A.; Vila, J.; Scheraga, H. A. *J. Comput. Chem.* **1991**, *12*(5), 594–605.

[41] Li, Z.; Scheraga, H. A. *Proc. Natl. Acad. Sci.* **1987**, *84*(19), 6611–6615.

[42] Wales, D. J.; Doye, J. P. K. *J. Phys. Chem. A* **1997**, *101*(28), 5111–5116.

[43] Alder, B. J.; Wainwright, T. E. *J. Chem. Phys.* **1959**, *31*, 459–466.

[44] Lemons, D. S.; Anthony, G. *Les Comptes Rendus de l'Académie des sciences* **1908**, *146*, 530–533.

[45] Schlick, T. *Molecular Modeling and Simulation: An Interdisciplinary Guide;* Springer-Verlag New York, **2010**.

[46] Ermak, D. L.; McCammon, J. A. *J. Chem. Phys.* **1978**, *69*(4), 1352–1360.

[47] Ermak, D. L.; McCammon, J. A. *J. Chem. Phys.* **1978**, *69*, 1352–1360.

[48] McCammon, J. A.; Harvey, S. C. *Dynamics of Proteins and Nucleic Acids;* Cambridge Univ Press, **1987**.

[49] Verlet, L. *Phys. Rev.* **1967**, *159*, 98–103.

[50] Hockeny, R. W. *Methods Comput. Phys.* **1970**, *9*.

[51] Potter, D. *Computational Physics;* Wiley, **1972**.

[52] Swope, W.; Andersen, H. C.; Berens, H.; Wilson, K. R. *J. Chem. Phys.* **1982**, *76*, 637–649.

[53] Woodcock, L. *Chem. Phys. Lett.* **1971**, *10*, 257–261.

[54] Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. *J. Chem. Phys.* **1984**, *81*, 3684–3690.

[55] Morishita, T. *J. Chem. Phys.* **2000**, *113*, 2976–2982.

[56] Andersen, H. C. *J. Chem. Phys.* **1980**, *72*, 2384–2393.

[57] Tanaka, H.; Nakanishi, K.; Watanabe, N. *J. Chem. Phys.* **1983**, *78*(5), 2626–2634.

[58] Nose, S. *J. Chem. Phys.* **1984**, *81*(1), 511–519.

[59] Hoover, W. G. Mar **1985**, *31*, 1695–1697.

[60] Clausius, R. *Annalen der Physik;* **1870**.

[61] Parrinello, M.; Rahman, A. *Phys. Rev. Lett.* **1980**, *45*, 1196–1199.

[62] Martyna, G. J.; Tobias, D. J.; Klein, M. L. *J. Chem. Phys.* **1994**, *101*, 4177–4189.

[63] Nose, S.; Klein, M. L. *Mol. Phys.* **1983**, *50*, 1055–1076.

[64] Feller, S. E.; Zhang, Y.; Pastor, R. W.; Brooks, B. R. *J. Chem. Phys.* **1995**, *103*, 10267–10276.

[65] Feller, S. E.; Zhang, Y.; Pastor, R. W.; Brooks, B. R. *J. Chem. Phys.* **1995**, *103*, 4613–4621.

[66] Evans, D. J.; Morriss, G. P. *Comput. Phys. Rep.* **1984**, *1*, 297–343.

[67] McDonald, I. R.; Singer, K. *J. Chem. Phys.* **1967**, *47*(11), 4766–4772.

[68] McDonald, I. R.; Singer, K. *Farad. Discuss.* **1967**, *43*(11), 40–49.

[69] Valleau, J. P.; Card, D. N. *J. Chem. Phys.* **1972**, *57*(12), 5457–5462.

[70] Bennett, C. H. *J. Comput. Phys.* **1976**, *22*(2), 245–268.

[71] Torrie, G. M.; Valleau, J. P. *Chem. Phys. Lett.* **1974**, *28*(4), 578–581.

[72] Torrie, G. M.; Valleau, J. P. *J. Comput. Phys.* **1977**, *23*(2), 187–199.

[73] Lee, J. K.; Barker, J. A.; Abraham, F. F. *J. Chem* **1973**, *58*(8), 3166–3180.

[74] Mruzik, M. R.; Abraham, F. F.; Schreiber, D. E.; Pound, G. M. *J. Chem. Phys.* **1976**, *64*(2), 481–491.

[75] McGinty, D. J. *J. Chem. Phys.* **1973**, *58*(11), 4733–4742.

[76] Kirkwood, J. G. *J. Chem. Phys.* **1935**, *3*, 300–313.

[77] Kirkwood, J. G. *Theory of Liquids;* Gordon and Breach: New York, **1968**.

[78] DeDonder, T. *Acad. Royale de Belgique* **1927**, *9*.

[79] Mezei, M.; Swaminathan, S.; Beveridge, D. L. *J. Am. Chem. Soc.* **1978**, *100*(10), 3255–3256.

[80] Mezei, M. *Mol. Phys.* **1982**, *47*(6), 1307–1315.

[81] Patey, G. N.; Valleau, J. P. *J. Chem. Phys.* **1975**, *63*(6), 2334–2339.

[82] Okazaki, S.; Nakanishi, K.; Touhara, H.; Adachi, Y. *J. Chem. Phys.* **1979**, *71*(6), 2421–2429.

[83] Pangali, C.; Rao, M.; Berne, B. J. *J. Chem. Phys.* **1979**, *71*(7), 2975–2981.

[84] Pratt, L. R.; Chandler, D. *J. Chem. Phys.* **1977**, *67*(8), 3683–3704.

[85] Postma, J. P. M.; Berendsen, H. J. C.; Haak, J. R. *Faraday Symp. Chem. Soc.* **1982**, *17*, 55–67.

[86] Lee, C. Y.; Scott, H. L. *J. Chem. Phys.* **1980**, *73*(9), 4591–4596.

[87] Warshel, A. *J. Phys. Chem.* **1982**, *86*(12), 2218–2224.

[88] Tembe, B. L.; McCammon, J. A. *Comp. Chem.* **1984**, *8*, 281–283.

[89] Jorgensen, W. L.; Briggs, J. M.; Gao, J. *J. Am. Chem. Soc.* **1987**, *109*(22), 6857–6858.

[90] Jorgensen, W. L.; Briggs, J. M. *J. Am. Chem. Soc.* **1989**, *111*(12), 4190–4197.

[91] Chandrasekhar, J.; Smith, S. F.; Jorgensen, W. L. *J. Am. Chem. Soc.* **1984**, *106*(10), 3049–3050.

[92] Tobias, D. J.; III, C. L. B. *Chem. Phys. Lett.* **1987**, *142*(6), 472–476.

[93] Bash, P. A.; Singh, U. C.; Langridge, R.; Kollman, P. A. *Science* **1987**, *236*(4801), 564–568.

[94] Bash, P. A.; Singh, U. C.; Brown, F. K.; Langridge, R.; Kollman, P. A. *Science* **1987**, *235*(4788), 574–576.

[95] Rao, S. N.; Singh, U. C.; Bash, P. A.; Kollman, P. A. *Nature* **1987**, *328*(6130), 551–554.

[96] Kumar, S.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A.; Rosenberg, J. M. *J. Comput. Chem.* **1992**, *13*(8), 1011–1021.

[97] Souaille, M.; Roux, B. *Comput. Phys. Commun.* **2001**, *135*(1), 40–57.

[98] Ferrenberg, A. M.; Swendsen, R. H. *Phys. Rev. Lett.* **1989**, *63*, 1195–1198.

[99] Bartels, C.; Karplus, M. *J. Comput. Chem.* **1997**, *18*(12), 1450–1462.

[100] Beutler, T. C.; Mark, A. E.; van Schaik, R. C.; Gerber, P. R.; van Gunsteren, W. F. *Chem. Phys. Lett.* **1994**, *222*(6), 529–539.

[101] Ewald, P. P. *Annalen der Physik* **1921**, *369*, 253–287.

[102] Hummer, G.; Pratt, L. R.; Garcia, A.; E., A. *J. Phys. Chem.* **1996**, *100*(4), 1206–1215.

[103] Boresch, S.; Karplus, M. *J. Chem. Phys.* **1996**, *105*(12), 5145–5154.

[104] Fixman, M. *Proc. Natl. Acad. Sci.* **1974**, *71*, 3050–3053.

[105] Gao, N.; Scheraga, H. A. *Macromolecules* **1976**, *9*(4), 535–542.

[106] Den Otter, W. K.; Briels, W. J. *J. Chem. Phys.* **1998**, *109*(11), 4139–4146.

[107] Den Otter, W. K.; Briels, W. J. *Mol. Phys.* **2000**, *98*(12), 773–781.

[108] Darve, E.; Pohorille, A. *J. Chem. Phys.* **2001**, *115*(20), 9169–9183.

[109] Darve, E.; Wilson, M. A.; Pohorille, A. *Mol. Simul.* **2002**, *28*(1-2), 113–144.

[110] Henin, J.; Chipot, C. *J. Chem. Phys.* **2004**, *121*(7), 2904–2914.

[111] Chipot, C.; Henin, J. *J. Chem. Phys.* **2005**, *123*(24), 2440906.

[112] Laio, A.; Parrinello, M. *Proc. Natl. Acad. Sci.* **2002**, *99*(20), 12562–12566.

[113] Jarzynski, C. *Phys. Rev. Lett.* **1997**, *78*, 2690–2693.

[114] Jarzynski, C. *Phys. Rev. E* **1997**, *56*, 5018–5035.

[115] Hendrix, D. A.; Jarzynski, C. *J. Chem. Phys.* **2001**, *114*(14), 5974–5981.

[116] Jensen, M. .; Park, S.; Tajkhorshid, E.; Schulten, K. *Proc. Natl. Acad. Sci.* **2002**, *99*(10), 6731–6736.

[117] Izrailev, S.; Stepaniants, S.; Isralewitz, B.; Kosztin, D.; Lu, H.; Molnar, F.; Wriggers, W.; Schulten, K. In *Computational Molecular Dynamics: Challenges, Methods, Ideas*, pages 39–65. Springer-Verlag, **1998**.

[118] Jorgensen, W. L.; Ruiz-Caro, J.; Tirado-Rives, J.; Basavapathruni, A.; Anderson, K. S.; Hamilton, A. D. *Bioorg. Med. Chem. Lett.* **2006**, *16*(3), 663–667.

[119] Pearlman, D. A.; Charifson, P. S. *J. Med. Chem.* **2001**, *44*(21), 3417–3423.

[120] Glover, F. *ORSA J. Comput.* **1989**, *1*, 190–206.

[121] Glover, F. *ORSA J. Comput.* **1990**, *2*, 4–32.

[122] Jensen, F. *Introduction to Computational Chemistry;* John Wiley & Sons, **2006**.

[123] Pulay, P. *Chem. Phys. Lett.* **1980**, *73*(2), 393 – 398.

[124] Capelle, K. *Braz. J. Phys.* **2006**, *36*, 1318–1343.

[125] Stewart, E. T. *J. Chem. Soc.* **1959**, pages 1856–1860.

[126] Stewart, J. J. P. *J. Comput. Chem.* **1991**, *12*, 320–341.

[127] Stewart, J. J. P. *J. Mol. Model.* **2007**, *13*, 1173–1213.

[128] Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. *J. Am. Chem. Soc.* **1985**, *107*, 3902–3909.

[129] Ponder, J. W.; Wu, C.; Ren, P.; Pande, V. S.; Chodera, J. D.; Schnieders, M. J.; Haque, I.; Mobley, D. L.; Lambrecht, D. S.; DiStasio, R.; Head-Gordon, M.; Clark, G. N. I.; Johnson, M. E.; Head-Gordon, T. *J. Phys. Chem. B* **2010**, *114*, 2549–2564.

[130] London, F. Z. *Physik* **1930**, *63*, 245.

[131] Lennard-Jones, J. E. *Proc. R. Soc. London. Ser. A* **1924**, *106*, 463.

[132] Halgren, T. A. *J. Am. Chem. Soc.* **1992**, *114*(20), 7827–7843.

[133] Applequist, J. *Chem. Phys.* **1984**, *85*, 279.

[134] Applequist, J. *J. Chem. Phys.* **1985**, *83*, 809.

[135] Applequist, J. *J. Phys. A: Math. Gen.* **1989**, *22*, 4303.

[136] Chipote, C.; Phorille, A. *Free energy calculations - Theory and Application in Chemistry and Biology;* Springer, **2007**.

[137] Hill, T. L. *An Introduction to Statistical Thermodynamics;* Dover: New York, **1986**.

[138] McQarrie, D. A. *Statistical Mechanics;* Harper and Row: New York, **1976**.

[139] Cam, L. L. *Statist. sci* **1986**, *1*, 78–91.

[140] Pearson, K. *Biometrika* **1924**, *16*, 402–404.

[141] Tabor, M. *Chaos and Integrability in Nonlinear Dynamics: An Introduction;* Wiley: New York, **1989**.

[142] Kästner, J. *Wires Comput. Mol. Sci.* **2011**, *1*(6), 932–942.

[143] Ferrenberg, A. M.; Swendsen, R. H. *Phys. Rev. Lett.* **1988**, *61*, 2635–2638.

[144] Grebner, C.; Becker, J.; Weber, D.; Bellinger, D.; Tafipolski, M.; Brückner, C.; Engels, B. *J. Comput. Chem.* **2014**, *35*(24), 1801–1807.

[145] MacKerell, A. D. *Abstracts of Papers of the American Chemical Society* **1998**, *216*, U696–U696.

[146] MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, W. E.; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiorkiewicz-Kuczera, J.; Yin, D.; Karplus, M. *J. Phys. Chem. B* **1998**, *102*(18), 3586–3616.

[147] Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, *117*(19), 5179–5197.

[148] Damm, W.; Frontera, A.; Tirado-Rives, J.; Jorgensen, W. L. *J. Comput. Chem.* **1997**, *18*(16), 1955–1970.

[149] Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1996**, *118*, 11225–11236.

[150] Tafipolsky, M.; Engels, B. *J. Chem. Theory Comput.* **2011**, *7*(6), 1791–1803.

[151] Ansorg, K.; Tafipolsky, M.; Engels, B. *J. Phys. Chem. B* **2013**, *117*(35), 10093–10102.

[152] Stewart, J. J. P. *Stewart Computational Chemistry MOPAC2012* **2012**.

[153] Mattson, W.; Rice, B. M. *Comput. Phys. Commun.* **1999**, *119*, 135–148.

[154] Heinz, T. N.; Hünenberger, P. H. *J. Comput. Chem.* **2004**, *25*(12), 1474–1486.

[155] Gonnet, P. *J. Comput. Chem.* **2007**, *28*(2), 570–573.

[156] Hwang, K. *Advanced Computer Architecture: Parallelism, Scalability, Programmability;* McGraw-Hill Higher Education, 1st ed., **1992**.

[157] OpenMP application program interface version 3.0. May **2008**.

[158] Deserno, M.; Holm, C. *J. Chem. Phys.* **1998**, *109*(18), 7678–7693.

[159] Darden, T.; Perera, L.; Li, L.; Pedersen, L. *Structure* **1999**, *7*(3), 55 – 60.

[160] Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. *J. Chem. Phys.* **1995**, *103*(19), 8577–8593.

[161] Darden, T.; York, D.; Pedersen, L. *J. Chem. Phys.* **1993**, *98*(12), 10089–10092.

[162] *"The Design and Implementation of FFTW3," by Matteo Frigo amd Steven G. Johnson, Invited paper*, **1999**.

[163] *"A Fast Fourier Transform Compiler," by Matteo Frigo, in the Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '99)*, **1999**.

[164] Beeman, D. *J. Comput. Phys.* **1976**, *20*(2), 130–139.

[165] Levitt, M.; Meirovitch, H.; Huber, R. *J. Mol. Biol.* **1983**, *168*(3), 617 – 620.

[166] Andersen, H. C. *J. Comput. Phys.* **1983**, *52*(1), 24–34.

[167] Forester, T. R.; Smith, W. *J. Comput. Chem.* **1998**, *19*(1), 102–111.

[168] Ufimtsev, I. S.; Martinez, T. J. *J. Chem. Theory Comput.* **2009**, *5*(10), 2619–2628.

[169] Kaestner, J.; Carr, J. M.; Keal, T. W.; Thiel, W.; Wander, A.; Sherwood, P. *J. Phys. Chem. A* **2009**, *113*(43), 11856–11865.

[170] Martyna, G. J.; Tuckerman, M. E.; Tobias, D. J.; Klein, M. L. *Mol. Phys.* **1996**, *87*(5), 1117–1157.

[171] Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. *J. Comput. Phys.* **1977**, *23*(3), 327–341.

[172] Gonnet, P. *J. Comput. Phys.* **2007**, *220*(2), 740–750.

[173] Hansen, J. P.; Levesque, D.; Weis, J. J. *Phys. Rev. Lett.* **1979**, *43*, 979–982.

[174] Kratky, K. *J. Comput. Phys.* **1980**, *37*(2), 205 – 217.

[175] Kratky, K.; Schreiner, W. *J. Comput. Phys.* **1982**, *47*(2), 313 – 320.

[176] Schreiner, W.; Kratky, K. W. *J. Chem. Soc. Faraday Trans. 2* **1982**, *78*, 379–389.

[177] de Souza, O. N.; Ornstein, R. L. *Biophys. J.* **1997**, *72*, 2395–2397.

[178] Kuzkin, V. *J. Appl. Math. Mech.* **2014**, pages 1–6.

[179] Cheatham, T. E. I.; Miller, J. L.; Fox, T.; Darden, T. A.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, *117*(14), 4193–4194.

[180] McNaught, A. D.; Wilkinson, A. *IUPAC. Compendium of Chemical Terminology, 2nd ed. (the "Gold Book");* Blackwell Scientific Publications, Oxford, **1997**.

[181] Bekker, H.; Berendsen, H. J. C.; van Gunsteren, W. F. *J. Comput. Chem.* **1995**, *16*(5), 527–533.

[182] Zwanzig, R. W. *J. Chem. Phys.* **1954**, *22*, 1420–1426.

[183] Jorgensen, W. L.; Ravimohan, C. *J. Chem. Phys.* **1985**, *83*(6), 3050–3054.

[184] Pearlman, D. A. *J. Phys. Chem.* **1994**, *98*(5), 1487–1493.

[185] Boresch, S.; Karplus, M. *J. Phys. Chem. A* **1999**, *103*(1), 103–118.

[186] Boresch, S.; Karplus, M. *J. Phys. Chem. A* **1999**, *103*(1), 119–136.

[187] Gao, J.; Kuczera, K.; Tidor, B.; Karplus, M. *Science* **1989**, *244*(4908), 1069–1072.

[188] Simonson, T. *Mol. Phys.* **1993**, *80*(2), 441–447.

[189] Resat, H.; Mezei, M. *J. Chem. Phys.* **1993**, *99*(8), 6052–6061.

[190] Pitera, J. W.; van Gunsteren, W. F. *Mol. Simul.* **2002**, *28*(1-2), 45–65.

[191] Lin, C.-L.; Wood, R. H. *J. Comput. Chem.* **1994**, *15*(2), 149–154.

[192] Steinbrecher, T.; Mobley, D. L.; Case, D. A. *J. Chem. Phys.* **2007**, *127*(21), 214108.

[193] Steinbrecher, T.; Joung, I.; Case, D. A. *J. Comput. Chem.* **2011**, *32*(15), 3253–3263.

[194] Zacharias, M.; Straatsma, T. P.; McCammon, J. A. *J. Chem. Phys.* **1994**, *100*(12), 9025–9031.

[195] Ponder, J. W.; Richards, F. M. *J. Comput. Chem.* **1987**, *8*, 1016–1024.

[196] Humphrey, W.; Dalke, A.; Schulten, K. *J. Mol. Graph.* **1996**, *14*, 33–38.

[197] Friesen, P. *Implementierung von impliziten Solvens-Modeel in CAST;* Bachelorthesis, University of Würzburg, **2014**.

[198] Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. *J. Comp. Chem.* **2004**, *25*, 1157–1174.

[199] Wang, J.; Wang, W.; A., K. P.; Case, D. A. *J. Mol. Graph. Model.* **2006**, *25*, 247260.

[200] Zoete, V.; Cuendet, M. A.; Grosdidier, A.; Michielin, O. *J. Comput. Chem.* **2011**, *32*(11), 2359–2368.

[201] Grebner, C.; Niebling, S.; Schmuck, C.; Schluecker, S.; Engels, B. *Mol. Phys.* **2013**, *111*(16-17), 2489–2500.

[202] G., H.; Jonsson, H. *J. Chem. Phys.* **1999**, *111*, 7010–7022.

[203] Heyden, A.; Bell, A. T.; Keil, F. J. *J. Chme. Phys.* **2005**, *123*, 224101.

[204] Kästner, J.; Sherwood, P. *J. Chem. Phys.* **2008**, *128*, 014106.

[205] Shang, C.; Liu, Z.-P. *J. Chem. Theory Comput.* **2010**, *6*, 1136–1144.

[206] Snir, M.; Otto, S.; Huss-Ledermann, S.; Walker, D.; Dongarra, J. *MPI: The Complete Reference;* MIT Press Cambridge: MA USA, **1995**.

[207] Grebner, C.; Becker, J.; Stepanenko, S.; Engels, B. *J. Comput. Chem.* **2011**, *32*(10), 2245–2253.

[208] Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*(2), 926–935.

[209] Berendsen, H. J.; Grigera, J. R.; Straatsma, T. P. *J. Phys. Chem.* **1987**, *91*(24), 6269–6271.

[210] Onufriev, A.; Bashford, D.; Case, D. *Proteins* **2004**, *55*(2), 383–394.

[211] Miertus, S.; Scrocco, E.; Tomasi, J. *Chem. Phys.* **1981**, *55*(1), 117–129.

[212] Miertus, S. S.; Tomasi, J. *Chem. Phys.* **1982**, *65*(2), 239–245.

[213] Klamt, A.; Schuurmann, G. *J. Chem. Soc. Perkin Trans. 2* **1993**, pages 799–805.

[214] Born, M. *Z. Phys.* **1920**, *1*, 45–48.

[215] Atkins, P. W.; MacDermott, A. J. *J. Chem. Edu.* **1982**, *59*(5), 359–360.

[216] Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. *J. Am. Chem. Soc.* **1990**, *112*(16), 6127–6129.

[217] Hermann, R. B. *J. Phys. Chem.* **1972**, *76*(19), 2754–2759.

[218] Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. *J. Am. Chem. Soc.* **1990**, *112*(16), 6127–6129.

[219] Onufriev, A.; Case, D. A.; Bashford, D. *J. Comput. Chem.* **2002**, *23*(14), 1297–1304.

[220] Onufriev, A.; Case, D. A.; Bashford, D. *J. Comput. Chem.* **2002**, *23*(14), 1297–1304.

[221] Richmond, T. J. *J. Mol. Biol.* **1984**, *178*(1), 63–89.

[222] Meyer, A. Y. *J. Comput. Chem.* **1988**, *9*(1), 18–24.

[223] Hasel, W.; Hendrickson, T. F.; Still, W. *Tetrahedron Comput. Methodol.* **1988**, *1*(2), 103 – 116.

[224] Qiu, D.; Shenkin, P.; Hollinger, F.; Still, W. *J. Phys. Chem. A* **1997**, *101*(16), 3005–3014.

[225] Hawkins, G. D.; Cramer, C. J.; Truhlar, D. G. *Chem. Phys. Lett.* **1995**, *246*(1-2), 122–129.

[226] Makarov, V.; Pettitt, B.; Feig, M. *Acc. Chem. Res.* **2002**, *35*(6), 376–384.

[227] Implementierung von impliziten solvens-methoden in cast. Friesen, P. **2014**.

[228] Hudson, P. S.; White, J. K.; Kearns, F. L.; Hodoscek, M.; Boresch, S.; Woodcok, H. L. *Biochim. Biohphys. Acta - General Subjects* **2015**, *1850*, 944–953.

[229] Allinger, N. L.; Grev, R. S.; Yates, B. F.; Schaefer III, H. F. *J. Am. Chem. Soc.* **1990**, *112*, 114–118.

[230] Mo, Y. *J. Org. Chem.* **2010**, *75*, 2733–2736.

[231] Murcko, M. A.; Castejon, H.; Wiberg, K. B. *J. Phys. Chem.* **1996**, *100*, 16162–16168.

[232] Levy, Y.; Jortner, J.; Becker, O. M. *Proc. Natl. Acad. Sci.* **2001**, *98*(5), 2188–2193.

[233] Park, S.; Khalili-Araghi, F.; Tajkhorshid, E.; Schulten, K. *J. Chem. Phys.* **2003**, *119*(6), 3559–3566.

[234] Marcus, Y. *J. Chem. Soc., Faraday Trans* **1991**, *87*(18), 2995–2999.

[235] Straatsma, T. P.; Berendsen, H. J. *J. Chem. Phys.* **1988**, *89*(9), 5876–5886.

[236] Hummer, G.; Garde, S.; Garcia, A.; Pohorille, A.; Pratt, L. *Proc. Natl. Acad. Sci.* **1996**, *93*(17), 8951–8955.

[237] Day, P.; Pachter, R.; Gordon, M.; Merrill, G. *J. Chem. Phys.* **2000**, *112*(5), 2063–2073.

[238] Maheshwary, S.; Patel, N.; Sathyamurthy, N.; Kulkarni, A.; Gadre, S. *J. Phys. Chem. A* **2001**, *105*(46), 10525–10537.

[239] Kazimirski, J. K.; Buch, V. *J. Phys. Chem. A* **2003**, *107*(46), 9762–9775.

[240] Kabrede, H. *Chem. Phys. Lett.* **2006**, *430*(4-6), 336–339.

[241] James, T.; Wales, D. J.; Hernandez Rojas, J. *J. Chem. Phys.* **2007**, *126*(5).

[242] Kazachenko, S.; Thakkar, A. J. *Chem. Phys. Lett.* **2009**, *476*(1-3), 120–124.

[243] Temelso, B.; Archer, K. A.; Shields, G. C. *J. Phys. Chem. A* **2011**, *115*(43), 12034–12046.

[244] Do, H.; Besley, N. A. *J. Chem. Phys.* **2012**, *137*(13).

[245] Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*(2), 926–935.

[246] Becke, A. D. *J. Chem. Phys.* **1993**, pages 5648–5652.

[247] Lee, C.; Yang, W.; Parr, R. G. *Phys. Rev. B* **1988**, pages 785–789.

[248] Vosko, S. H.; Wilk, L.; Nusair, M. *Can. J. Phys.* **1980**, pages 1200–1211.

[249] Stephens, P. J.; Devlin, F. J. Chabalowski, C. F.; Frisch, M. J. *J. Phys. Chem. B* **1994**, pages 11623–11627.

[250] Grimme, S. *J. Comput. Chem.* **2006**, *27*, 1787–1799.

[251] Hehre, W. J.; Ditchfield, R.; Pople, J. A. *J. Chem. Phys.* **1972**, *56*(5), 2257–2261.

[252] Ditchfield, R.; Hehre, W. J.; Pople, J. A. *J. Chem. Phys.* **1971**, *54*, 724–728.

[253] Hariharan, P. C.; Pople, J. A. *Mol. Phys.* **1974**, *27*, 209–214.

[254] Gordon, M. S. *Chem. Phys. Lett.* **1980**, *76*, 163–168.

[255] Francl, M. M.; Pietro, W. J.; Hehre, W. J.; Binkley, J. S.; J., D. D.; Pople, J. A.; Gordon, M. S. *J. Chem. Phys.* **1982**, *77*, 3654–3665.

[256] Binning Jr, R. C.; Curtiss, L. A. *J. Comput. Chem.* **1990**, *11*, 1206–1216.

[257] Blaudeau, J.-P.; McGrath, M. P.; Curtiss, L. A.; Radom, L. *J. Chem. Phys.* **1997**, *107*, 5016–5021.

[258] Rassolov, V. A.; Pople, J. A.; Ratner, M. A.; Windus, T. L. *J. Chem. Phys.* **1983**, *78*, 1223–1229.

[259] Rassolov, V. A.; Ratner, M. A.; Pople, J. A.; Redfern, P. C.; Curtiss, L. A. *J. Comput. Chem.* **2201**, *22*, 976–984.

[260] Ramya, L.; Gautham, N. *Biopolymers* **2012**, *97*(3), 165–176.

[261] Chen, M.; Cuendet, M. A.; Tuckerman, M. E. *J. Chem. Phys.* **2012**, *137*(2).

[262] Tyka, M. D.; Jung, K.; Baker, D. *J. Comput. Chem.* **2012**, *33*(31), 2483+.

[263] Shen, M.; Freed, K. *Biophys. J.* **2002**, *82*(4), 1791–1808.

[264] Klepeis, J.; Pieja, M.; Floudas, C. *Biophys. J.* **2003**, *84*(2, 1), 869–882.

[265] Wales, D. J.; Bogdan, T. V. *J. Phys. Chem. B* **2006**, *110*(42), 20765–20776.

[266] Zhan, L.; Chen, J. Z. Y.; Liu, W.-K. *Biophys. J.* **2006**, *91*(7), 2399–2404.

[267] Dirac, P. A. M. *Proc. Royal. Soc. A* **1929**, *123*, 714–733.

[268] Slater, J. C. *Phys. Rev.* **1951**, *81*, 385–390.

[269] Becke, A. D. *Phys. Rev. A* **1988**, *38*, 3098–3100.

[270] Dekker, M. *Handbook of Conducting Polymers;* CRC Press, **1998**.

[271] Würthner, F. *Chem. Comm.* **2004**, pages 1564–1579.

[272] Oswald, P.; Würthner, F. *J. Am. Chem. Soc.* **2007**, *129*, 14319–14326.

[273] Chen, Z.; Debije, M. G.; Debaerdemaeker, T.; Osswald, P.; Würthner, F. *Chem. Phys. Chem.* **2004**, *5*, 137–140.

[274] Shi, Y.; Qian, H.; Li, Y.; Yue, W.; Wang, Z. *Org. Lett.* **2008**, *10*, 740–743.

[275] Yong, Y.; Lam, J. W. Y.; Tang, B. Z. *Chem. Soc. Rev.* **2011**, *40*, 5361–5388.

[276] Yokoyama, D. *J. Mater. Chem.* **2011**, *21*, 19187–19202.

[277] Wang, C.; Dong, H.; Hu, W.; Liu, Y.; Zhu, D. *Chem. Rev.* **2012**, *112*, 2208–2267.

[278] Johns, J. E.; Muller, E. A.; Frechet, J. M.; Haarris, C. B. *J. Am. Chem. Soc.* **2010**, *132*, 15720–15725.

[279] Jimenez, A. J.; Lin, M.-J.; Burschka, C.; Becker, J.; Settels, V.; Engels, B.; Wurthner, F. *Chem. Sci.* **2014**, *5*, 608–619.

[280] Kerr, I. D.; Lee, J. H.; Farady, C. J.; Marion, R.; Rickert, M.; Sajid, M.; Pandey, C. R.; Legac, J.; Hansell, E.; McKerrow, J. H.; Craik, C. S.; Rosenthal, P. J.; Brinen, L. S. *J. Biol. Chem.* **2009**, *284*, 25697Ű25703.

[281] Laufer, M. K.; Djimde, A. A.; Plowe, C. V. *Am. J. Trop. Med. Hyg.* **2007**, *77*(6, S), 160–169.

[282] Ouellette, M. *Trop. Med. Int. Health* **2001**, *6*(11), 874–882.

[283] Caffrey, C. R.; Hansell, E.; Lucas, K. D.; Brinen, L. S.; Hernandez, A. A.; Cheng, J.; II, S. L. G.; Roush, W. R.; Stierhof, Y.-D.; Bogyo, M.; Steverding, D.; McKerrow, J. H. *Mol. Biol. Para.* **2001**, *118*(1), 61–73.

[284] Nkemgu, N. J.; Grande, R.; Hansell, E.; McKerrow, J. H.; Caffrey, C. R.; Steverding, D. *Int. J. Antimicrob. Agents* **2003**, *22*(2), 155–159.

[285] Eakin, A. E.; Mills, A. A.; Harth, G.; McKerrow, J. H.; Craik, C. S. *J. Biol. Chem.* **1992**, *267*(11), 7411–7420.

[286] Engel, J. C.; Doyle, P. S.; McKerrow, J. H. *Med. B. Aires* **1999**, *59*(2), 171–175.

[287] Vicik, R.; Hoerr, V.; Glaser, M.; Schultheis, M.; Hansell, E.; McKerrow, J. H.; Holzgrabe, U.; Caffrey, C. R.; Ponte-Sucre, A.; Moll, H.; Stich, A.; Schirmeister, T. *Bioorg. Med. Chem. Lett.* **2006**, *16*(10), 2753–2757.

[288] Ettari, R.; Nizi, E.; Di Francesco, M. E.; Dude, M.-A.; Pradel, G.; Vicik, R.; Schirmeister, T.; Micale, N.; Grasso, S.; Zappalaă, M. *J. Med. Chem.* **2008**, *51*(4), 988–996.

[289] Jaishankar, P.; Hansell, E.; Zhao, D.-M.; Doyle, P. S.; McKerrow, J. H.; Renslo, A. R. *Bioorg. Med. Chem. Lett.* **2008**, *18*(2), 624–628.

[290] Palmer, J. T.; Rasnick, D.; Klaus, J. L.; Bromme, D. *J. Med. Chem.* **1995**, *38*(17), 3193–3196.

[291] Rosenthal, P. J.; Olson, J. E.; Lee, G. K.; Palmer, J. T.; Klaus, J. L.; Rasnick, D. *Antimicrob. Agents Chemother.* **1996**, *40*(7), 1600–1603.

[292] Shenai, B. R.; Lee, B. J.; Alvarez-Hernandez, A.; Chong, P. Y.; Emal, C. D.; Neitz, R. J.; Roush, W. R.; Rosenthal, P. J. *Antimicrob. Agents Chemother.* **2003**, *47*(1), 154–160.

[293] Lavergne, S. N.; Park, B. K.; Naisbitt, D. J. *Curr. Opin. Allergy Clin. Immunol.* **2008**, *8*, 299–307.

[294] Park, B. K.; Sanderson, J. P.; Naisbitt, D. J. *Drugs as haptens, antigens, and immunogens.;* Karger, Basel; Switzerland, **2007**.

[295] Uetrecht, J. *Chem. Res. Toxicol.* **2008**, *21*, 84–92.

[296] Uetrecht, J. *Chem. Res. Toxicol.* **2009**, *22*, 24–34.

[297] M., M.; Fink, R. F.; Thiel, W.; Schirmeister, T.; Engels, B. *J. Am. Chem. Soc.* **2008**, *130*(27), 8696–8705.

[298] Liu, P.; D., F.; Cai, W.; Chipot, C. *J. Chem. Theory Comput.* **2012**, *8*(8), 2606–2616.

[299] Pohorille, A.; Jarzynski, C.; Chipot, C. *J. Phys. Chem. B* **2010**, *114*(32), 10235–10253.

[300] Lee, C. Y.; Scott, H. L. *J. Chem. Phys.* **1980**, *73*(9), 4591–4596.

[301] Shirts, M. R.; Bair, E.; Hooker, G.; Pande, V. S. *Phys. Rev. Lett.* **2003**, *91*(14), 140601.

[302] Dennis, J. E. J.; Schnabel, R. B. *SIAM Rev.* **1979**, *4*, 443–459.

[303] Hildebrand, F. B. *Introduction to numerical analysis;* Dover Publications, Inc., New York, **1987**.

[304] Ortega, J. M.; Rheinboldt, W. C. *Iterative solution of nonlinear equations in several variables;* Academic Press, New York-London, **1970**.

[305] TURBOMOLE V7.0 2015, a development of University of Karlsruhe and Forschungszentrum Karlsruhe GmbH, 1989-2007, TURBOMOLE GmbH, since 2007; available from
`http://www.turbomole.com`.

[306] Dunning Jr, T. H. *J. Chem. Phys.* **1989**, *90*, 1007–1023.

[307] Kendall, R. A.; Dunning Jr, T. H.; Harrison, R. J. *J. Chem. Phys.* **1992**, *96*, 6796–6806.

[308] Peterson, K. A.; Woon, D. E.; Dunning Jr, T. H. *J. Chem. Phys.* **1994**, *100*, 7410–7415.

[309] Onsager, L. *J. Am. Chem. Soc.* **1936**, *58*(8), 1486–1493.

[310] Weast, R. C. *Handbook of Chemistry and Physics;* CRC Press: Boca Raton, FL, **1987**.

[311] Warshel, A.; Aqvist, J. *Annu. Rev. Biophys. Biophys. Chem.* **1991**, *20*(1), 267–298.

[312] Simonson, . T.; Brooks, C. L. I. *J. Am. Chem. Soc.* **1996**, *118*(35), 8452–8458.

[313] Baker, N. A.; Sept, D.; Joseph, S.; Holst, M. J.; McCammon, J. A. *Proc. Natl. Acad. Sci.* **2001**, *98*(18), 10037–10041.

A. Appendix

A.1. CAST Results

The following tables show the CAST results for the calculations as performed in Section 5.4.2 for vacuum and solvated systems.

A.1.1. Vacuum

	H⟷Br	H⟷F	Br⟷F	Cycle
Run1	26.10	28.36	2.06	-0.20
Run2	26.53	27.80	2.38	1.11
Run3	26.72	28.84	2.14	0.02
Average	**26.45 ± 0.26**	**28.33 ± 0.42**	**2.19 ± 0.13**	**0.31 ± 0.57**

Table A.1.: Simulation results for the single transformations in vacuum and averages of the three runs. The transformation of $H \longleftrightarrow F$ yields a slightly higher free energy change than for the $H \longleftrightarrow Br$ transformation. The difference is recovered for the $Br \longleftrightarrow F$ transformation. Energies are given in kcal/mol.

A.1.2. Solvent

	H⟷Br	H⟷F	Br⟷F	Cycle
Run1	31.57	33.99	1.82	-0.60
Run2	30.79	33.81	2.03	-0.99
Run3	30.41	33.29	1.52	-1.36
Average	**30.92 ± 0.48**	**33.69 ± 0.29**	**1.79 ± 0.21**	**-0.98 ± 0.31**

Table A.2.: Simulation results for the single transformations in solvent and averages of the three runs. The transformation of $H \longleftrightarrow F$ yields a slightly higher free energy change than for the $H \longleftrightarrow Br$ transformation. The difference is recovered for the $Br \longleftrightarrow F$ transformation. Energies are given in kcal/mol.

Eklärung

Hiermit erkläre ich an Eides statt, dass ich die Dissertation

Development and Implementation of new simulation possibilities in the CAST program package

selbständig angefertigt und keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt habe.

Ich erkläre außerdem, dass diese Dissertation weder in gleicher oder anderer Form bereits in einem anderen Prüfungsverfahren vorgelegen hat.

Ich habe früher außer den mit dem Zulassungsgesuch urkundlich vorgelegten Graden keine weiteren akademischen Grade erworben oder zu erwerben versucht.

Würzburg, den 28.11.2015 _____