Algorithms and software for efficient biomolecular simulation

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH

for the degree of
Doctor of Natural Sciences

presented by
MARKUS CHRISTEN
Dipl. Chem. ETH
born March 29, 1975
citizen of Bülach (ZH) and Wynau (BE), Switzerland

accepted on the recommendation of
Prof. Dr. Wilfred F. van Gunsteren, examiner
Prof. Dr. Andrew E. Torda and Prof. Dr. Matthias Troyer, co-examiners

2006
Acknowledgements

It was with great pleasure that I spent my last five years with the group of IGC. This is for no small reason due to Wilfred, who, with his never ending enthusiasm, patience (especially when correcting the many sign errors I did - which may cancel if you just do enough of them) and humour, created a warm and friendly atmosphere in his group. His confidence and trust in me (be it concerning GROMOS, teaching, or the mere fact that I would finally produce a thesis), never got me bored. Together with Jolande, they were cheerful hosts the many times we were invited. Thank you.

Thanks also go to my co-examiners Andrew Torda and Matthias Troyer for reading this thesis and allowing me to defend it. Especially, I thank Andrew for the discussion about mixing fine-grained and coarse-grained simulations which brought me along towards Chapter 5.

I started working in this group a long time ago. In this time I enjoyed working together with many different people. Especially I remember: Xavier (for writing a Burg-Arras talk on the spot), Tomas (for his views on any topic), Heiko (for cleaning my apartment), Salomon (for his genius), Urs (for sleeping on the floor while awaiting a meeting with Wilfred), Roland (for leaving me with a mysterious server and now providing me with a - hopefully less mysterious - job), Dirk (for having to take the backseat in a Fiat cinquecento), Lukas (for showing me all about butane), Christian (for teaching me Informatik), Fred (for believing I quit and made a career taking care of elderly people), Jed (for not complaining about my endless discussions with Tomas), Indira (for taking care of us all), Regula (for her nice smile), Phil (for drinking a bowl of sangria after an already funny borrel) and Janez (for showing me Ljubljana).

Of course, most important were the people of my own generation that were here during most of my time within the group: Thereza and Roberto and Phaedra (because they introduced us to wine-tasting parties), Christine (because of her ability to see the funny side of everything), David (because of a nice game of tennis), Nico (because of inviting us all to Mainz), Alex (because everybody has his groundhog days), Alice (because she stayed sysadmin with me in good and in bad times), Haibo (because he never succumbed to the access of evil), Peter (because he just got the hang of everything), Mika (despite his attitude problem), Ulf (for his dry humour), Vincent (because he upheld the ideas of Mao), Michel (because of him showing us the desert) Tim (because of his teasing of Zrinka and sharing the last months with me) and especially Chris (because of all the 10 minutes of work we shared - e.g. “ene_ana”).
During the last months (and years), I enjoyed the presence of Daan (because he is helpful in every way), Valerie (because of sharing BiOMOS), Zrinka (because she keeps the group healthy with propolis), Jozi and Urban (because they bring joy to my home), Merijn (because of “it’s Friday” discussions) and Annemarie (because I, too, like deep snow skiing), Bojan (because his guitar rules), Maria-Grazia (because of her one-of-a-kind tiramisu and her vivid opinions about politicians), Riccardo (because of his cooperativity and entropy), Cristina (because she brought Brazilian sunshine to us), Daniel (because “you know, it works for me”), Moritz (because of his Austrian charm), Claire (because of her French way of life), Lovorka (because of her refreshing attitude), Vreni (for sharing the GROMOS05 burden) and Maria (because of letting the Suns shine).

Almost all of the work presented in this thesis relies heavily on MD05 (GROMOSXX for us). I could never have written all of it alone. Some of the preparatory work was already started by Heiko, and brought to the next level by Roland with GROMOS++. But most of the long nights discussing design, new features or just hunting a tricky bug were shared with Chris who contributed a lot not only to the eventual success but also to a very pleasurable experience of getting there.

After Chris left Daniel took over that burden, but in addition we enjoyed the airport of Athens during the Olympic Games, wake boarding and sailing under the worst possible weather conditions and a lot of beach volley. This definitely brightens daily life.

Some of the most productive times were the months spent working together with semester students. Clara was the first one and she had therefore a bit of a rough time with finding and fixing bugs in the, at that time very fresh, program code. Without her I do not think GROMOSXX would have reached maturity. I am happy that we could finally write Chapter 9. Next to follow was Bettina. With her, the program was already much more stable, but as compensation the scientific problem seemed to be a bit evasive. Still, we somehow managed to get to Chapter 7, which is one of my favourite chapters of this thesis. During the last semester-project I was working with Pitschna. We were struggling to rotate rings, to bend helices and to successfully put glucose into chair conformations. This work resulted in Chapter 6. I had a lot of fun working on all these projects.

When I started Prisca made sure that everything went in a smooth way. After the move to Hönggerberg Daniela took over and I am very grateful for her ease in handling group matters and keeping Wilfred organized as well as for her friendship.

Obviously, a thesis is not written in one day but a rather longish process, building up on things one has done before. All my life I have been happy to receive the trust and support from my family: my parents Christina and Felice, my brothers Stefan and Daniel and my grandparents. Thank you. I also greatly appreciated the cordiality and easy acceptance I received as an additional member of the Rosenberger family.

Most important, especially also during this quite stressful last few months was the continuous support from Bea. Without her managing our household and often also my life next to a demand-
ing study and job of her own while still patiently listening to incomprehensible statistical mechanics problems getting this far would have been all but impossible. Even more important was the time we spent without thinking about work.

I would like to dedicate this thesis to her.
Contents

Acknowledgements i
Kurzfassung ix
Summary xi
Publications xiii

1 Introduction 1
   1.1 Force field ........................................ 5
      1.1.1 Bonded interaction terms ...................... 5
      1.1.2 Non-bonded interaction terms .................. 8
   1.2 Equations of motion ................................ 9
   1.3 Biased sampling and free energy determination .... 11
   1.4 Deterministic chaos ................................ 15
   1.5 Outline ........................................... 18
   1.6 Bibliography ...................................... 18

2 The GROMOS software for biomolecular simulation: GROMOS05 27
   2.1 Summary ............................................ 27
   2.2 Introduction ...................................... 28
   2.3 Overview of functionalities ....................... 29
   2.4 Algorithms ........................................ 31
      2.4.1 MD algorithm .................................. 31
      2.4.2 New features .................................. 32
   2.5 Code organisation, implementation ................ 53
      2.5.1 MD engine in FORTRAN: PROMD ................. 53
      2.5.2 MD engine in C++: MD++ ........................ 54
      2.5.3 Analysis modules: GROMOS++ ................. 62
   2.6 Examples of application ........................... 65
## Contents

2.6.1 Local-elevation simulation of glucose ........................................... 65
2.6.2 Replica-exchange simulation of butane ........................................ 66
2.6.3 Coarse-grained simulation of alkanes ......................................... 66
2.6.4 One-step perturbation calculations on the free energy of ligand binding to the estrogen receptor ........................................... 71
2.6.5 Other applications ................................................................. 71
2.7 Conclusions ................................................................................. 73
2.8 Acknowledgements .................................................................... 73
2.9 Bibliography ............................................................................... 74

3 On searching in, sampling of, and dynamically moving through conformational space of biomolecular systems: a review 87
3.1 Summary .................................................................................... 87
3.2 Introduction ................................................................................ 88
3.3 Choice of degrees of freedom ....................................................... 91
3.4 Types of searching methods ......................................................... 93
3.5 Techniques to speed up a simulation ............................................. 94
3.6 Search and sampling enhancement techniques ............................ 96
3.7 Biasing the search, sampling or simulation ................................... 101
3.8 Sampling or simulation along pathways ....................................... 101
3.9 Use of other than spatial molecular coordinates when searching or sampling ................................................................. 102
3.10 Discussion .................................................................................. 103
3.11 Acknowledgements .................................................................... 104
3.12 Bibliography ............................................................................. 104

4 Investigation of sampling efficiency using configurational entropy as a measure 119
4.1 Summary .................................................................................... 119
4.2 Introduction ................................................................................ 120
4.3 Method ....................................................................................... 120
4.4 Model ........................................................................................ 122
4.5 Results ....................................................................................... 122
4.6 Discussion .................................................................................. 129
4.7 Acknowledgements .................................................................... 130
4.8 Bibliography ............................................................................. 133

5 Multigraining: an algorithm for simultaneous fine-grained and coarse-grained simulation of molecular systems 135
5.1 Summary .................................................................................... 135
# Contents

5.2 Introduction .................................................................................................................. 136
5.3 Method .......................................................................................................................... 136
  5.3.1 Multi-graining Hamiltonian using mapping of coarse-grained particles onto fine-grained ones .............................................................................................................. 136
  5.3.2 Replica-exchange multigraining simulation ................................................................ 141
  5.3.3 Multi-graining Hamiltonian with partial mapping of coarse-grained particles onto fine-grained ones ................................................................................................. 142
5.4 Results .......................................................................................................................... 143
5.5 Discussion ...................................................................................................................... 148
5.6 Acknowledgements ........................................................................................................ 149
5.7 Bibliography .................................................................................................................. 150

6 Sampling of rare events using hidden restraints ................................................................ 153
  6.1 Summary ....................................................................................................................... 153
  6.2 Introduction ................................................................................................................... 154
  6.3 Method ........................................................................................................................... 155
    6.3.1 Distance restraints .................................................................................................. 157
    6.3.2 Dihedral-angle restraints ....................................................................................... 158
  6.4 Applications .................................................................................................................. 159
    6.4.1 Conformations of a cyclic aminoxy-hexapeptide upon binding cations and anions ......................................................................................................................... 160
    6.4.2 Relative stabilities of hexopyranose in $^4C_1$ vs. $^1C_4$ conformation .................. 161
  6.5 Discussion ..................................................................................................................... 168
  6.6 Appendix ...................................................................................................................... 169
    6.6.1 Distance constraints .............................................................................................. 170
    6.6.2 Dihedral-angle constraints .................................................................................... 172
  6.7 Acknowledgments ......................................................................................................... 176
  6.8 Bibliography ................................................................................................................ 176

7 Adaptive restraints using local-elevation simulation .......................................................... 181
  7.1 Summary ....................................................................................................................... 181
  7.2 Introduction ................................................................................................................... 182
  7.3 Theory ........................................................................................................................... 183
  7.4 Methods and results ...................................................................................................... 184
  7.5 Discussion ..................................................................................................................... 191
  7.6 Acknowledgements ....................................................................................................... 192
  7.7 Bibliography ................................................................................................................ 195
## 8 Approximate flexible distance constraints

8.1 Summary ........................................................................................................................................ 199
8.2 Introduction .................................................................................................................................. 200
8.3 Hard constraints ............................................................................................................................ 200
8.4 Flexible constraints ....................................................................................................................... 201
8.5 Numerical experiments .................................................................................................................. 205
8.5.1 Ethane collision in the gas phase ............................................................................................... 207
8.5.2 Neopentane liquid .................................................................................................................... 208
8.6 Discussion ....................................................................................................................................... 213
8.7 Acknowledgements ...................................................................................................................... 216
8.8 Appendix ....................................................................................................................................... 216
8.9 Bibliography .................................................................................................................................. 223

## 9 Free energy calculations using flexible-constrained, hard-constrained and non-constrained MD simulations

9.1 Summary ....................................................................................................................................... 225
9.2 Introduction .................................................................................................................................... 226
9.3 Method ........................................................................................................................................... 226
9.4 Molecular models and computational procedure ........................................................................... 232
9.5 Results ........................................................................................................................................... 234
9.6 Conclusion ...................................................................................................................................... 237
9.7 Acknowledgements ...................................................................................................................... 239
9.8 Bibliography .................................................................................................................................. 240

## 10 Outlook

10.1 Bibliography ............................................................................................................................... 245

Curriculum Vitae .................................................................................................................................. 247
Kurzfassung


Ein Verfahren, um eine fein-körnige (atomistische) und eine grob-körnige Representation eines Systems zugleich zu simulieren wird in Kapitel 5 gegeben. Der momentane Zustand kann
Kurzfassung


Zum Schluss wird in Kapitel 10 auf mögliche zukünftige Entwicklungen von Simulationsprogrammen und effizienten Algorithmen hingewiesen.
Summary

Over the last fifty years, continuous development of simulation algorithms and software has made classical molecular dynamics simulation into a powerful tool to investigate biomolecular processes in atomistic detail. After a short introduction of classical simulation with special attention to biased sampling of configurational space and some techniques to calculate free energies, the latest version of the Groningen Molecular Simulation package is introduced: GROMOS05. In this version, the heart of the package is delivered in two variants. An enhanced version of PROMD, the FORTRAN simulation engine, and MD05, written in C++. MD05 strives for higher modularity and readability by making use of object oriented features and generic programming techniques. All algorithms presented in the other chapters are integrated into MD05.

A brief overview and classification of searching methods is given in Chapter 3 before a closer look is taken at entropies calculated by replica-exchange stochastic dynamics simulations. For the simple test system complete sampling of configurational space can be achieved. Therefore the entropies calculated from standard stochastic dynamics simulations should match the ones obtained from the replica-exchange simulations. This was found to be true. At low temperature and incomplete sampling of the configurational space, replica-exchange simulation can be more efficient if the simulation parameters are carefully selected. The procedure of rotationally fitting structures to decompose entropy into configurational, translational and rotational entropies has a significant impact on this decomposition, often disfavouring configurational entropy at low temperatures.

An algorithm to combine fine-grained (atomistic) simulations with coarse-grained ones is introduced in Chapter 5. The algorithm allows to either continuously vary the grain-level of the simulation from fully fine-grained to coarse-grained and back, or, using replica exchange, simultaneously simulate a system at fine-grained, at coarse-grained and at some intermediate grain levels. The higher the grain level the bigger the simulation time-step may be, therefore increasing sampling efficiency. Through replica exchange, the replicas at lower grain-level can profit from the faster sampling available at the higher levels.

A different way to improve sampling is shown in the next chapter, where the simulation is restrained to follow a reaction coordinate from a given state A to a state B for obtaining the free energy difference between these two states. Opposed to standard methods, the restraining
functions are chosen to have zero potential energy and forces in the end states (A and B). This permits the calculation of a path independent free energy difference. The method, implemented in terms of distance and dihedral restraints is applied to ion-complexation in a cyclic peptide and the calculation of the relative stability of two different chair conformations of a glucopyranoside. For the second case a comparison with a potential of mean force calculated using dihedral-angle constraints is given.

Further on, restraints can be used to force a simulation to reproduce experimental data. If the force field is not well suited to represent a biomolecule or if the simulation time is not long enough to sample the relevant part of the configurational space, experimental properties might not be reproduced by simulations. Both problems may be overcome by adding restraints. In Chapter 7 adaptive restraints based on local-elevation simulation are introduced. Adaptive restraints have two main advantages: First, as the restraint force slowly builds up over the simulation time, and only if the restraint is not fulfilled, a minimum of force is added, which leads to the least possible disturbance of the simulation. Second, as the adaptive restraint is based on local-elevation simulation, locally enhanced sampling is achieved, until the restraint is fulfilled.

In the next two chapters the focus is no longer on restraints but on constraints. But these are flexibilized using an approximate, but fast algorithm, which enables simulations to use a comparatively large time-step but still have the constraints slowly adapt to changes in the environment. Using this technique, fast bond vibrational frequencies can be avoided. The method is applied to neopentane simulations under high pressure and to a comparative study of different models of flexibility. In this study, the free energy difference of water and methanol is calculated by thermodynamic integration.

Finally, in Chapter 10 an outlook of future development with regard to simulation software and configurational sampling is provided.
Publications

This thesis has led to the following publications:

**Chapter 2:**
Markus Christen, Philippe H. Hünenberger, Dirk Bakowies, Riccardo Baron, Roland Bürgi, Daan P. Geerke, Tim N. Heinz, Mika A. Kastenholz, Vincent Kräutler, Chris Oostenbrink, Christine Peter, Daniel Trzesniak, and Wilfred F. van Gunsteren,
"The GROMOS Software for Biomolecular Simulation: GROMOS05"

**Chapter 3:**
Markus Christen and Wilfred F. van Gunsteren,
"On searching in, sampling of, and dynamically moving through conformational space of biomolecular systems: a review"

**Chapter 5:**
Markus Christen and Wilfred F. van Gunsteren,
"Multigraining: an algorithm for simultaneous fine-grained and coarse-grained simulation of molecular systems"

**Chapter 6:**
Markus Christen, Anna-Pitschna E. Kunz, and Wilfred F. van Gunsteren,
"Sampling rare events using hidden restraints"
Chapter 8:
Markus Christen and Wilfred F. van Gunsteren,
"An approximate but fast method to impose flexible distance constraints in molecular dynamics simulations"
Related publications:

Wilfred F. van Gunsteren, Dirk Bakowies, Roland Bürgi, Indira Chandrasekhar, Markus Christen, Xavier Daura, Peter Gee, Alice Glättli, Tomas Hansson, Chris Oostenbrink, Christine Peter, Jed Pitera, Lukas Schuler, Thereza Soares, and Haibo Yu,
“Molecular dynamics simulation of biomolecular systems”
*CHIMIA* 55 (2001), 856–860

Thereza A. Soares, Markus Christen, Kai F. Hu, and Wilfred F. van Gunsteren,
“Alpha- and beta-polypeptides show a different stability of helical secondary structure”
*Tetrahedron* 60 (2004), 7775–7780

Wilfred F. van Gunsteren, Dirk Bakowies, Riccardo Baron, Indira Chandrasekhar, Markus Christen, Xavier Daura, Peter Gee, Daan P. Geerke, Alice Glättli, Philippe H. Hünenberger, Mika A. Kastenholz, Chris Oostenbrink, Merijn Schenk, Daniel Trzesniak, Nico F. A. van der Vegt, and Haibo B. Yu,
“Biomolecular modelling: goals, problems perspectives”

Bettina Keller, Markus Christen, Chris Oostenbrink, and Wilfred F. van Gunsteren,
“On using oscillating time-dependent restraints in MD simulation”
*J. Biomol. NMR* (2006), accepted
Chapter 1

Introduction

*It is a metaphysical doctrine that from the same antecedents follow the same consequents. No one can deny this. But it is not much use in a world like this, in which the same antecedents never again concur, and nothing ever happens twice.*

— James Clark Maxwell

Only a limited number of properties of a biomolecular system is accessible to experimental measurement. Over fifty years ago, this limitation lead to the formulation of a general theoretical method of calculating the properties of any substance which may be considered as composed of \( N \) interacting individual particles, on a fast electronic computing machine\(^1\). Any equilibrium property of interest \( < Q > \) may be calculated by evaluating the phase-space integral of the corresponding microscopic observable \( q(r, p) \)

\[
< Q > = \frac{\int q(r, p) \exp\left(-\mathcal{H}(r, p)/k_BT\right) dr dp}{\int \exp\left(-\mathcal{H}(r, p)/k_BT\right) dr dp} \tag{1.1}
\]

with \( \mathcal{H}(r, p) = \mathcal{K}(p) + \mathcal{V}(r) \) the Hamiltonian of the system, using the potential energy of the system \( \mathcal{V}(r) \), depending on particle positions \( r \), and the kinetic energy of the system, depending on the conjugate momenta \( p \) of the particles, where \( r \) and \( p \) represent \( 3N \) dimensional vectors and \( k_B \) is Boltzmann's constant. For the 3 dimensional vector of the position and the momentum of particle \( i \), \( r_i \) and \( p_i \) will be used, respectively. As it is evidently impractical to carry out this integral for more than a couple of particles using standard numerical integration techniques, the Monte Carlo (MC) method\(^2\) was applied. The Monte Carlo method fills the gap between classical mechanics (of few particles) using ordinary differential equations and statistical mechanics (of very many particles) using the theory of probabilities. By studying a great, but by no means
Chapter 1. Introduction

exhaustive, number of possible states \( \{ r, p \} \) chosen at random, the average value \( < Q > \) of property \( Q \) may be determined, with probability and accuracy dependent on the number of states considered. The computational procedure is the following: Given an ensemble of particles with positions \( r(n) \) and momenta \( p(n) \), random processes are initiated which lead to new states \( \{ r(n + 1), p(n + 1) \} \). Repeating the procedure \( N_{mc} \) times leads to an ensemble of states, according to given probability distributions for each of the random processes. The essential feature of the method is that it avoids dealing with multiple integrations, but instead samples a single chain of events. The generated states can be used in statistical studies of a property \( Q \)

\[
< Q > \approx \frac{1}{N_{mc}} \sum_{n=1}^{N_{mc}} q(r(n), p(n)).
\]

Comparison to Equation 1.1 shows, that instead of generating all states (or a number of random states) and weighting those by \( \exp(-\mathcal{H}(r(n), p(n))/k_B T) \), here, the states are weighted evenly. Therefore, they need to occur with a probability proportional to \( \exp(-\mathcal{H}(r(n), p(n))/k_B T) \) in the generated ensemble of states. This may be achieved by introducing an acceptance criterion for each move from state \( \{ r(n), p(n) \} \) to \( \{ r(n + 1), p(n + 1) \} \) based on the potential energy difference \( \Delta E = \mathcal{V}(r(n + 1)) - \mathcal{V}(r(n)) \). If \( \Delta E < 0 \), i.e., if the move would bring the system to a state of lower energy, it is allowed. Otherwise, the move is only allowed with a probability of \( \exp(-\Delta E/k_B T) \) and the total kinetic energy \( \mathcal{K}(p) \) of the system at a given temperature is constant. This conditional acceptance of a (Monte Carlo) move is usually referred to as the Metropolis criterion.

In the Monte Carlo method the moves of the particles are artificial. Therefore only the average of a property \( Q \) is meaningful, not the time-series of the microscopic observable \( q(r, p) \). Exchanging the artificial Monte Carlo moves by an integration of the classical equations of motion (see Section 1.2) leads to physical moves of the particles in a molecular dynamics (MD) simulation. Thermodynamic averages of a property \( Q \) can be estimated in molecular dynamics simulations by taking advantage of the identity of the ensemble average with the time average over an infinite period (given quasi-ergodicity of the system) and by approximating the infinite time average by the average over a finite period \( T \) obtained from a simulation:

\[
< Q > \approx \frac{1}{T} \int_0^T q(r(t), p(t))dt.
\]

Reaching long enough simulation times \( T \) to converge the average using molecular dynamics simulations is often problematic. Therefore, the convergence behaviour of any calculated property has to be investigated.

First molecular dynamics simulations were done almost fifty years ago, using hard spheres, later elastic spheres, but several years passed until a Lennard-Jones liquid and liquid water were simulated. In 1977, for the first time a protein was simulated using the molecular dynamics method, initiating the field of biomolecular simulation and changing the rigid picture of proteins into one of dynamic motion.
Over the years, tremendous progress has been made with respect to simulation methods, system sizes and simulation time lengths. Developing a highly accurate force field (the interaction potential energy function \( V(r) \)) is extremely difficult. First, the potential energy of a system is the sum over a huge number of individual terms. To achieve overall high accuracy, the accuracy of the individual terms needs to be orders of magnitude higher. Second, force-field development has to take into account entropic effects to produce a meaningful ensemble of states at non-zero temperature. But even with a perfect force field, the problem of searching or sampling the energy hypersurface, of which the dimensionality is given by the number of degrees of freedom present in the system, remains. The motions along these degrees of freedom show a variety of characteristics, from highly harmonic to anharmonic, chaotic and diffusive. Moreover, correlations are present that cover a wide range of time- and spatial scales, from femtoseconds and tenths of nanometers to milliseconds and micrometers. The energy hypersurface is therefore a very rugged surface. This makes the search for the global energy minimum, or rather the search for those regions that contribute most to the free energy of the system, a daunting if not impossible task. To alleviate this problem, a huge amount of methods exists to bias the sampling towards interesting regions of the (free) energy hypersurface (see Section 1.3).

Quite apart from the aforementioned problems, there is still the difficulty of comparing properties obtained from simulations to experimental data. An experimental measurement of a quantity \( Q \) yields an average \( \langle Q \rangle \) of a distribution over molecules and time. Upon the averaging, the detailed information on the distribution is lost, and very different distributions may yield the same average. The average of property \( Q \) as given in Equation 1.1 is the expectation value of the corresponding microscopic observable \( q(r,p) \), in other words the integral over the phase-space of \( q(r,p) \) multiplied by the probability of the state \( \{r,p\} \), which is given, for the canonical ensemble at a volume \( V \) and a temperature \( T \), by

\[
P_{NVT}(r,p) = \frac{\exp(-\mathcal{H}(r,p)/k_BT)}{\int \exp(-\mathcal{H}(r,p)/k_BT) dr dp} = \frac{\exp(-\mathcal{H}(r,p)/k_BT)}{h^3N!Z(N,V,T)},
\]

where \( h \) is Planck’s constant and \( Z(N,V,T) \) is the canonical partition function, defined as

\[
Z(N,V,T) = \frac{1}{h^3N!} \int \exp(-\mathcal{H}(r,p)/k_BT) dr dp,
\]

and the factor \( N! \) is only present in case the particles are indistinguishable. In the experimentally important isothermic-isobaric ensemble the partition function is given as

\[
Z(N,p,T) = \frac{1}{Vh^3N!} \int \exp(-(\mathcal{H}(r,p) + pV)/k_BT) dr dp dV
\]

and the corresponding phase-space probability also depends on the volume

\[
P_{NpT}(r,p) = \frac{\exp(-(\mathcal{H}(r,p) + pV)/k_BT)}{Vh^3N!Z(N,p,T)}.
\]
Straightforward integration of the equations of motion in molecular dynamics simulations leads to microcanonical ensembles. Extensions of the formalism exist to carry out simulations in the canonical or isothermic-isobaric ensemble as well (see Section 2.4.2, “Thermostat algorithms” and “Barostat algorithms”). The instantaneous pressure in a molecular dynamics simulation is computed as

\[ \mathcal{P} = \frac{1}{2} V^{-1}(\mathcal{K} - \mathcal{W}) \]  

(1.8)

where

\[ \mathcal{K} = \frac{1}{2} \sum_{i=1}^{N} m_i^{-1} p \otimes p, \]  

(1.9)

and

\[ \mathcal{W} = \frac{3}{2} V \frac{\partial V(r)}{\partial V} \]  

(1.10)

are the instantaneous kinetic energy and virial tensors. In the special case of a pairwise-additive interaction term \( V_p(r) \), the virial contribution is

\[ \mathcal{W}_p = -\frac{1}{2} \sum_{i}^{N} \sum_{j>i}^{N} f_{p,ij} \otimes r_{ij} \]  

(1.11)

with \( r_{ij} = r_i - r_j \) and \( f_{p,ij} \) the pairwise force from interaction term \( V_p(r) \) exerted by atom \( j \) on atom \( i \).

Whenever simulations and experiments are compared, the results of these comparisons may be classified as follows. If agreement between simulation and experiment is obtained, this may be due to the following reasons: The simulation adequately reflects the experimental system. Or, the property examined is insensitive to the details of the simulation. Variation of the simulation parameters would not change the agreement. Or, a compensation of errors has occurred. If only a few, global or system properties for a system with very many degrees of freedom are calculated and compared, this situation can easily emerge. If no agreement between simulation and experiment is obtained, this may be due to one or both of the following reasons: The simulation does not reflect the experimental system. The theory or model is incorrect, or the simulated property is not converged, or the software is at fault or incorrectly used. Or, the experimental data are incorrect. Therefore, comparisons of simulation results with experimental data have to be carefully analysed not to draw any wrong conclusions.

In the next section, the heart of a molecular dynamics simulation, the force field and the equations of motion, will be introduced, followed by a brief description of various methods to bias sampling of conformational space and finally by an explanation of methods for the determination of free energy differences using molecular dynamics simulations.
1.1 Force field

A force field used in classical molecular dynamics simulations specifies the functional form of \( V(r) \) and also the (most often empirical) parameters used to model a given (biomolecular) system. Together with the GROMOS software package the GROMOS force field has been developed since the early 1980's, when a first set of (non-bonded) parameters was specified\(^{25} \). Since then, the force field has been improved continuously\(^{26-32} \). The most recent refinement\(^{31}\) was based primarily on reproducing the free enthalpies of hydration and apolar solvation for a range of small compounds, and led to the parameter sets 53A5 (for pure liquids) and 53A6 (in aqueous solution).

To specify the functional form of the GROMOS force field, it is common to distinguish physical (atomic) interactions and special (non-physical) interactions:

\[
V(r) = V_{\text{phys}}(r) + V_{\text{special}}(r).
\]

(1.12)

The special interactions include, among others, restraints applied to the system (see Section 1.3). The physical interaction terms themselves can be divided into the bonded and the non-bonded interactions,

\[
V_{\text{phys}}(r) = V_{\text{bon}}(r) + V_{\text{nonbon}}(r),
\]

(1.13)

where the bonded interactions are the sum of bond, bond-angle, harmonic (improper) dihedral angle and trigonometric (torsional) dihedral angle terms. The non-bonded interactions are the sum of van der Waals (Lennard Jones) and electrostatic (Coulomb with Reaction Field) interactions between (in principle) all pairs of atoms.

In the following sections, these interaction terms are defined and the corresponding expressions for the force on a particle \( i \)

\[
f_i = -\frac{\partial}{\partial r_i} V'(r)
\]

(1.14)

and the virial contribution (according to Equation 1.10) are given.

1.1.1 Bonded interaction terms

Covalent bond interaction

For the covalent bond interaction, two different functional forms are present in GROMOS. The standard functional form is quartic:

\[
V_{\text{bond}}(r) = \sum_{k=1}^{N_b} \frac{1}{4} K_{b_k} (b_k^2 - b_0^2)^2.
\]

(1.15)

The actual bond length of the \( k^{th} \) bond between atoms \( k_1 \) and \( k_2 \) with positions \( r_{k_1} \) and \( r_{k_2} \) is given by \( b_k = |r_{k_1k_2}| = \sqrt{r_{k_1k_2} \cdot r_{k_1k_2}} \) where \( r_{k_1k_2} = r_{k_1} - r_{k_2} \). \( K_{b_k} \) is the force constant and \( b_0 \) the
ideal length of bond $k$. The forces on atom $k_1$ and $k_2$ from bond $k$ are

$$f_{k_1}^{\text{bond}} = -K_{bk} (b_k^2 - b_{k_0}^2) \mathbf{r}_{k_1k_2}$$
$$f_{k_2}^{\text{bond}} = -K_{bk} (b_k^2 - b_{k_0}^2) \mathbf{r}_{k_2k_1}$$

(1.16)

(1.17)

The (atomic) virial contribution from bond $k$ is

$$W_k^{\text{bond}} = \frac{1}{2} f_{k_1}^{\text{bond}} \otimes \mathbf{r}_{k_1k_2}.$$  

(1.18)

Alternatively, a harmonic functional form may be used for the covalent bond terms:

$$q_{\text{harm bond}}^{\text{bond}}(\mathbf{r}) = \sum_{k=1}^{N_\theta} \frac{1}{2} K_{bk}^{\text{harm}} (b_k - b_{k_0})^2,$$

(1.19)

with the harmonic bond force constant $K_{bk}^{\text{harm}}$, and the harmonic forces from bond $k$ on the atoms $k_1$ and $k_2$

$$f_{k_1}^{\text{harm bond}} = -K_{bk}^{\text{harm}} (b_k - b_{k_0}) \frac{\mathbf{r}_{k_1k_2}}{|\mathbf{r}_{k_1k_2}|}$$
$$f_{k_2}^{\text{harm bond}} = -K_{bk}^{\text{harm}} (b_k - b_{k_0}) \frac{\mathbf{r}_{k_2k_1}}{|\mathbf{r}_{k_2k_1}|}.$$

(1.20)

(1.21)

The (atomic) virial contribution remains unchanged:

$$W_k^{\text{bond}} = \frac{1}{2} f_{k_1}^{\text{harm bond}} \otimes \mathbf{r}_{k_1k_2}.$$  

(1.22)

**Covalent bond-angle interaction**

The covalent bond-angle bending interaction reads

$$q_{\text{angle}}(\mathbf{r}) = \sum_{k=1}^{N_\theta} \frac{1}{2} K_{\theta k} (\cos \theta_k - \cos \theta_{k_0})^2,$$

(1.23)

where the bond-angle $k$ is defined by the atoms $k_1$, $k_2$ and $k_3$ and given by

$$\theta_k = \arccos \left( \frac{\mathbf{r}_{k_1k_2} \cdot \mathbf{r}_{k_3k_2}}{|\mathbf{r}_{k_1k_2}| |\mathbf{r}_{k_3k_2}|} \right).$$

(1.24)

The forces from bond-angle $k$ are

$$f_{k_1}^{\text{angle}} = -K_{\theta k} (\cos \theta_k - \cos \theta_{k_0}) \left( \frac{\mathbf{r}_{k_3k_2}}{|\mathbf{r}_{k_3k_2}|} - \frac{\mathbf{r}_{k_1k_2}}{|\mathbf{r}_{k_1k_2}|} \cos \theta_k \right) \frac{1}{|\mathbf{r}_{k_1k_2}|}$$
$$f_{k_3}^{\text{angle}} = -K_{\theta k} (\cos \theta_k - \cos \theta_{k_0}) \left( \frac{\mathbf{r}_{k_3k_2}}{|\mathbf{r}_{k_3k_2}|} - \frac{\mathbf{r}_{k_1k_2}}{|\mathbf{r}_{k_1k_2}|} \cos \theta_k \right) \frac{1}{|\mathbf{r}_{k_3k_2}|}$$
$$f_{k_2}^{\text{angle}} = -f_{k_1}^{\text{angle}} + f_{k_3}^{\text{angle}},$$

(1.25)

(1.26)

(1.27)

and the (atomic) virial contribution is

$$W_k^{\text{angle}} = \frac{1}{2} \left( f_{k_1}^{\text{angle}} \otimes \mathbf{r}_{k_1k_2} + f_{k_3}^{\text{angle}} \otimes \mathbf{r}_{k_3k_2} \right).$$

(1.28)
1.1. Force field

covalent harmonic (improper) dihedral-angle interaction

The term in the interaction function that represents the harmonic, so-called improper (out-of-plane, out-of-tetrahedral configuration) dihedral-angle bending interaction reads

$$q_{\text{improper}}(\mathbf{r}) = \sum_{k=1}^{N_C} \frac{1}{2} K_{k} (\zeta_k - \zeta_{0k})^2,$$  \hspace{1cm} (1.29)

with the (improper) dihedral-angle $k$ defined by the atoms $k_1, k_2, k_3$ and $k_4$, and $\zeta_k$ as

$$\zeta_k = \text{sign}(\zeta_k) \arccos \left( \frac{\mathbf{r}_{mj} \cdot \mathbf{r}_{nk}}{|\mathbf{r}_{mj}| |\mathbf{r}_{nk}|} \right)$$  \hspace{1cm} (1.30)

with

$$\mathbf{r}_{mj} = \mathbf{r}_{k_1 k_2} \times \mathbf{r}_{k_3 k_2},$$ \hspace{1cm} (1.31)

$$\mathbf{r}_{nk} = \mathbf{r}_{k_3 k_4} \times \mathbf{r}_{k_3 k_4},$$ \hspace{1cm} (1.32)

$$\text{sign}(\zeta_k) = \text{sign}(\mathbf{r}_{k_3 k_2} \cdot \mathbf{r}_{nk}).$$ \hspace{1cm} (1.33)

The forces from the improper dihedral-angle $k$ on atoms $k_1, k_2, k_3$ and $k_4$ are given as

$$f_{k_1}^{\text{improper}} = -K_{k} (\zeta_k - \zeta_{0k}) \frac{\mathbf{r}_{k_1 k_2}}{|\mathbf{r}_{mj}|} \mathbf{r}_{mj},$$ \hspace{1cm} (1.34)

$$f_{k_2}^{\text{improper}} = +K_{k} (\zeta_k - \zeta_{0k}) \frac{\mathbf{r}_{k_3 k_2}}{|\mathbf{r}_{nk}|} \mathbf{r}_{nk},$$ \hspace{1cm} (1.35)

$$f_{k_3}^{\text{improper}} = \left( \frac{\mathbf{r}_{k_1 k_2} \cdot \mathbf{r}_{k_3 k_2}}{|\mathbf{r}_{k_1 k_2}|^2} - 1 \right) \mathbf{f}_{k_1} - \frac{\mathbf{r}_{k_3 k_4} \cdot \mathbf{r}_{k_3 k_2}}{|\mathbf{r}_{k_3 k_2}|^2} \mathbf{f}_{k_4},$$ \hspace{1cm} (1.36)

$$f_{k_4}^{\text{improper}} = -\mathbf{f}_{k_1} - \mathbf{f}_{k_2} - \mathbf{f}_{k_4},$$ \hspace{1cm} (1.37)

and its (atomic) virial contribution is

$$q_{k}^{\text{improper}} = \frac{1}{2} \left( \mathbf{f}_{k_1} \otimes \mathbf{r}_{k_1 k_2} + \mathbf{f}_{k_3} \otimes \mathbf{r}_{k_3 k_2} + \mathbf{f}_{k_4} \otimes \mathbf{r}_{k_4 k_2} \right).$$ \hspace{1cm} (1.38)

covalent trigonometric (proper) dihedral-angle torsion interaction

The term in the interaction function that represents the trigonometric dihedral-angle torsion interaction reads

$$q_{\text{dihedral}}(\mathbf{r}) = \sum_{k=1}^{N_\delta} K_{\delta_k} \left( 1 + \cos(\delta_k) \cos(m_k \phi_k) \right),$$ \hspace{1cm} (1.39)

with parameters $\delta_k = 0$ or $\pi$ and $m_k = 1, 2, \ldots, 6$. The torsion angle $\phi_k$ is given as

$$\phi_k = \text{sign}(\phi_k) \arccos \left( \frac{\mathbf{r}_{im} \cdot \mathbf{r}_{in}}{|\mathbf{r}_{im}| |\mathbf{r}_{in}|} \right),$$ \hspace{1cm} (1.40)
where

\[
\begin{align*}
\mathbf{r}_{im} &= \mathbf{r}_{k_1 k_2} - \mathbf{r}_{k_1 k_2} \cdot \mathbf{r}_{k_3 k_2} \mathbf{r}_{k_3 k_2}, \\
\mathbf{r}_{ln} &= -\mathbf{r}_{k_1 k_4} + \mathbf{r}_{k_1 k_4} \cdot \mathbf{r}_{k_3 k_2} \mathbf{r}_{k_3 k_2}, \\
\text{sign}(\phi_k) &= \text{sign} \left( \mathbf{r}_{k_1 k_3} \cdot (\mathbf{r}_{k_3 k_2} \times \mathbf{r}_{k_3 k_4}) \right).
\end{align*}
\]

The forces from the dihedral-angle torsion \( k \) on atoms \( k_1, k_2, k_3 \) and \( k_4 \) are

\[
\begin{align*}
\mathbf{f}^\text{dihedral}_{k_1} &= -K \phi_k \cos \delta_k \frac{\partial \cos(m_k \phi_k)}{\partial \cos \phi_k} \left( \frac{\mathbf{r}_{ln}}{|\mathbf{r}_{ln}|} - \frac{\mathbf{r}_{im}}{|\mathbf{r}_{im}|} \cos \phi_k \right) \frac{1}{|\mathbf{r}_{im}|}, \\
\mathbf{f}^\text{dihedral}_{k_2} &= -K \phi_k \cos \delta_k \frac{\partial \cos(m_k \phi_k)}{\partial \cos \phi_k} \left( \frac{\mathbf{r}_{im}}{|\mathbf{r}_{im}|} - \frac{\mathbf{r}_{ln}}{|\mathbf{r}_{ln}|} \cos \phi_k \right) \frac{1}{|\mathbf{r}_{ln}|}, \\
\mathbf{f}^\text{dihedral}_{k_3} &= \left( \mathbf{r}_{k_1 k_2} \cdot \mathbf{r}_{k_3 k_2} \right) \mathbf{f}_{k_1} - \mathbf{r}_{k_3 k_2} \mathbf{f}_{k_4}, \\
\mathbf{f}^\text{dihedral}_{k_4} &= -\mathbf{f}_{k_1} - \mathbf{f}_{k_2} - \mathbf{f}_{k_4},
\end{align*}
\]

and \( \frac{\partial \cos(m_k \phi_k)}{\partial \cos \phi_k} \) is tabulated for the possible values of \( m_k \). The (atomic) virial contribution from the dihedral-angle torsion \( k \) is

\[
W^\text{dihedral}_k = \frac{1}{2} \left( \mathbf{f}^\text{dihedral}_{k_1} \otimes \mathbf{r}_{k_1 k_2} + \mathbf{f}^\text{dihedral}_{k_2} \otimes \mathbf{r}_{k_3 k_2} + \mathbf{f}^\text{dihedral}_{k_3} \otimes \mathbf{r}_{k_3 k_2} + \mathbf{f}^\text{dihedral}_{k_4} \otimes \mathbf{r}_{k_3 k_2} \right).
\]

### 1.1.2 Non-bonded interaction terms

The GROMOS force field has been parametrised using a pair-wise Lennard-Jones type interaction function for the van der Waals interactions

\[
q^{LJ}(\mathbf{r}) = \sum_{\text{pairs } i,j} \frac{C_{12ij}}{r_{ij}^{12}} - \frac{C_{6ij}}{r_{ij}^{6}},
\]

with \( C_{12ij} \) and \( C_{6ij} \) dependent on parameters \( C_{12k} \) and \( C_{6k} \) specified for all types of atoms, using geometric combination rules

\[
C_{12ij} = \sqrt{C_{12i} C_{12j}},
\]

\[
C_{6ij} = \sqrt{C_{6i} C_{6j}},
\]

and a Coulombic term representing electrostatic interactions

\[
q^E(\mathbf{r}) = \sum_{\text{pairs } i,j} \frac{q_i q_j}{4\pi \varepsilon_0 \varepsilon_1 r_{ij}}.
\]
where \( \varepsilon_0 \) is the dielectric permittivity of vacuum, \( \varepsilon_1 \) the relative permittivity of the medium in which the atoms are embedded and \( q_k \) the partial charges of the atoms. The value of \( \varepsilon_1 \) is standardly set to 1. In addition to the direct Coulombic interactions, a reaction-field contribution \( V_{r}\) the electrostatic interactions may be calculated, representing the interaction of atom \( i \) with the induced field of a continuous dielectric medium outside a cutoff distance \( R_{rf} \) due to the presence of atom \( j \):

\[
q_{rf}(r) = \sum_{\text{pairs } i,j} \frac{q_i q_j}{4\pi \varepsilon_0 \varepsilon_1} \frac{-1}{2} C_{rf} \frac{r_{ij}^2}{R_{rf}^3},
\]

where

\[
C_{rf} = \frac{(2\varepsilon_1 - 2\varepsilon_2)(1 + \kappa R_{rf}) - \varepsilon_2(\kappa R_{rf})^2}{(\varepsilon_1 + 2\varepsilon_2)(1 + \kappa R_{rf}) + \varepsilon_2(\kappa R_{rf})^2}
\]

and \( \varepsilon_2 \) and \( \kappa \) are the relative permittivity and inverse Debye screening length of the medium outside the cutoff sphere defined by \( R_{rf} \), respectively. And finally, a distance-independent reaction-field contribution, which ensures that the electrostatic energy is zero for atoms separated by a distance equal to the cutoff distance \( R_{rf} \):

\[
q_{rf} = \sum_{\text{pairs } i,j} \frac{q_i q_j}{4\pi \varepsilon_0 \varepsilon_1} \frac{-1}{2} C_{rf} \frac{r_{ij}^2}{R_{rf}^3}.
\]

The forces from the non-bonded interaction term \( q_{nbon}(r) = V_{r}^{LJ} + V_{r}^{C} + V_{r}^{rf} + V_{r}^{rf c} \) on atoms \( i \) and \( j \) are

\[
f_{i}^{\text{nbon}} = \left( \frac{12 C_{12ij}}{r_{ij}^4} - \frac{6 C_{6ij}}{r_{ij}^6} + \frac{q_i q_j}{4\pi \varepsilon_0 \varepsilon_1} \left( \frac{1}{r_{ij}^3} + \frac{C_{rf}}{R_{rf}^3} \right) \right) r_{ij},
\]

\[
f_{j}^{\text{nbon}} = -f_{i}^{\text{nbon}},
\]

and the virial contribution from the atom pair \( i, j \) is

\[
W_{ij}^{nbon} = \frac{1}{2} f_{i}^{nbon} \otimes r_{ij}.
\]

### 1.2 Equations of motion

If a system of \( N \) particles is represented by the (time-independent) Hamiltonian function

\[
\mathcal{H}(\mathbf{r}, \mathbf{p}) = \mathcal{K}(\mathbf{p}) + \mathcal{V}(\mathbf{r}),
\]

where \( \mathcal{K}(\mathbf{p}) = \sum_{i=1}^{N} p_i^2 / 2m_i \) is the kinetic energy of this system and \( m_i \) the mass of particle \( i \), its equations of motion are given as

\[
\frac{d}{dt} r_i = \frac{\partial \mathcal{H}(\mathbf{r}, \mathbf{p})}{\partial p_i},
\]

\[
\frac{d}{dt} p_i = -\frac{\partial \mathcal{H}(\mathbf{r}, \mathbf{p})}{\partial r_i}.
\]
In the special case of a Cartesian coordinate system, Equations 1.60 and 1.61 correspond to Newton’s equations of motion:

\[
\frac{d}{dt} r_i = m_i^{-1} p_i \\
\frac{d}{dt} p_i = f_i = -\frac{\partial}{\partial r_i} \mathcal{V}(r).
\]

As the Hamiltonian function \( \mathcal{H}(r, p) \) represents the total energy of the system and (in its time-independent form) its time-derivative is zero:

\[
\frac{d}{dt} \mathcal{H}(r, p) = \frac{d}{dt} (\mathcal{K}(p) + \mathcal{V}(r)) = 0,
\]

the total energy of the system is conserved.

To devise a numerical integration scheme for the equations of motion, several approaches may be employed. One common technique develops solutions using the Taylor series expansion of the positions and momenta at \( t + \Delta t \) about \( t \). Careful consideration of the resulting functional form allows the time reversal symmetry of the equations of motion to be preserved and an accuracy of second order in \( \Delta t \). A different approach based on an evolution operator formulation of classical mechanics was formulated in the last decade, also shown in a recent review. The equations of motion may be cast in the general form:

\[
\frac{d}{dt} x = i\mathcal{L} x
\]

where \( x \) is the phase space vector \( \{r, p\} \) and \( i\mathcal{L} \) is the Liouville operator given by:

\[
i\mathcal{L} = \{..., \mathcal{H}\} = \sum_{i=1}^{N} \left( \frac{\partial H}{\partial p_i} \frac{\partial}{\partial r_i} - \frac{\partial H}{\partial r_i} \frac{\partial}{\partial p_i} \right) = \sum_{i=1}^{N} \left( \frac{p_i}{m_i} \frac{\partial}{\partial r_i} + f_i \frac{\partial}{\partial p_i} \right).
\]

Equations of type 1.65 have the formal solution:

\[
x(t) = \exp(i\mathcal{L}t)x(0),
\]

which is the starting point for the derivation of numerical integration procedures. The unitary operator \( \exp(i\mathcal{L}t) \) is the classical propagator and its exponential is defined as a series expansion:

\[
\exp(i\mathcal{L}t) = 1 + i\mathcal{L}t - \frac{1}{2} \mathcal{L}^2 t^2 + ...
\]

If the classical propagator can be rewritten as the sum of two parts, \( i\mathcal{L} = i\mathcal{L}_1 + i\mathcal{L}_2 \), such that the action on \( x(0) \) can be evaluated analytically for each part, practical numerical integrators can be generated. This may be achieved by the following procedure. First, the classical propagator can be rewritten using the Trotter theorem into:

\[
\exp(i\mathcal{L}t) = \lim_{P \to \infty} \left( \exp \left( \frac{i\mathcal{L}_2 t}{2P} \right) \exp \left( \frac{i\mathcal{L}_1 t}{P} \right) \exp \left( \frac{i\mathcal{L}_2 t}{2P} \right) \right)^P.
\]
1.3. Biased sampling and free energy determination

Then defining $t/P = \Delta t$ allows the approximations

\[
\exp(i\hat{L}\Delta t) \approx \exp(i\hat{L}_2\Delta t/2)\exp(i\hat{L}_1\Delta t)\exp(i\hat{L}_2\Delta t/2) + O(\Delta t^3)
\] (1.70)

\[
\exp(i\hat{P}\Delta t) \approx \prod_{k=1}^{P} \exp(i\hat{L}_2\Delta t/2)\exp(i\hat{L}_1\Delta t)\exp(i\hat{L}_2\Delta t/2) + O(\Delta t^2).
\] (1.71)

Finally, with the obvious choices

\[
i\hat{L}_1 = \sum_{i=1}^{N} \frac{p_i}{m_i} \frac{\partial}{\partial r_i},
\]

\[
i\hat{L}_2 = \sum_{i=1}^{N} f_i \frac{\partial}{\partial p_i},
\] (1.72)

analytical evaluation of the individual parts of the propagator is possible and the velocity Verlet\(^{39}\) integrator is obtained\(^{15}\):

\[
r_i(t + \Delta t) = r_i(t) + \Delta t v_i(t) + \frac{\Delta t^2}{2m_i} f_i(t),
\] (1.73)

\[
v_i(t + \Delta t) = v_i(t) + \frac{\Delta t}{2m_i} (f_i(t) + f_i(t + \Delta t)),
\] (1.74)

with the velocity of particle $i$ given as $v_i = p_i/m_i$. This approach can be easily extended to treat systems with multiple time scales of motion, as has been shown in methods using reference system propagation algorithms (RESPA\(^{37}\)).

The leap-frog integrator\(^{35}\)

\[
r_i(t + \Delta t) = r_i(t) + v_i(t + \frac{1}{2} \Delta t)\Delta t + O(\Delta t^3),
\] (1.75)

\[
v_i(t + \frac{1}{2} \Delta t) = v_i(t - \frac{1}{2} \Delta t) + m_i^{-1} f_i \Delta t + O(\Delta t^3),
\] (1.76)

can be shown to be equivalent to the velocity Verlet algorithm\(^{40,41}\). As the velocities are known only at time points intermediate to those where the positions (and thus the potential energy) are known, they have to be recalculated through

\[
v_i(t) = \frac{1}{2} \left( v_i(t - \frac{1}{2} \Delta t) + v_i(t + \frac{1}{2} \Delta t) \right) + \frac{1}{16} (f_i(t - \Delta t) - f_i(t + \Delta t)) \frac{\Delta t}{m_i} + O(\Delta t^4).
\] (1.77)

Usually, only the first term of the series is used, as this one is readily available during simulation.

1.3 Biased sampling and free energy determination

Complete conformational sampling of the phase-space of biomolecules is almost never feasible, especially as many properties depend upon a balance between specific intramolecular solute - solute and intermolecular solute - solvent interactions. Then explicit solvent molecules
are required, increasing the computational costs of a simulation tremendously, due to the much increased number of interaction pairs, and due to the need of additional averaging over many solvent configurations.

From Equation 1.1 and Equation 1.4 follows that the expectation value $< Q >$ of a property $Q$ is, in the canonical ensemble, given by

$$< Q > = \int q(r,p)P_{NVT}(r,p) \, dr \, dp. \quad (1.78)$$

This implies that there are regions of the phase-space which contribute more to the average of $Q$ than others, according to their probability $P_{NVT}(r,p)$, unless there are very many states with low probability (of which the statistical-mechanical definition of the entropy $S = k_B \ln(W)$ provides a measure: $S = k_B \ln(W)$ with $W$ being the thermodynamic probability of the macro-state, i.e. the number of micro-states by which the macro-state may be realised). Therefore, it seems possible to devise methods to sample only the more relevant parts of the vast phase-space, thereby increasing the efficiency of the simulation. From the various possible ways to enhance searching or sampling of phase-space, only methods using molecular dynamics will be discussed here.

First, it is important to determine the degrees of freedom necessary to be present explicitly during the simulation. If it is possible to treat subsets of the degrees of freedom present in an effective, averaged way, the cost of a simulation might be reduced significantly. If molecules or molecular fragments can be treated as single particles or beads, whose motion is simulated using a simple force field describing inter-bead interactions with a smooth and short-ranged interaction energy function, the efficiency can be orders of magnitude higher than that of corresponding atomistic level simulations, at the expense of the loss of atomic detail and some accuracy. Such a reduction of degrees of freedom from an atomistic level model is called coarse-graining. Details of the implementation of a recently proposed coarse-grained model are given in Section 2.4.2, "Coarse-grained simulation".

Often, it may be of interest to fix a subset of degrees of freedom to specified values and only evolve the remaining degrees of freedom according to the equations of motion. This constraining of $N_c$ degrees of freedom can be introduced into the Hamiltonian of a system using Lagrange's method of undetermined multipliers

$$\mathcal{H}(r,p) = K(p) + V(r) + \sum_{k=1}^{N_c} \lambda_k g_k(r) \quad (1.79)$$

with

$$g_k(r) \equiv q_k(r) - q_0^k = 0, \quad (1.80)$$

where for holonomic constraints the microscopic observable $q_k$ only depends on the particle positions $r$ and $q_0^k$ is the ideal constraint value. This additional term in the Hamiltonian leads to
the constraint forces

$$f^{\text{constr}}_i = - \sum_{k=1}^{N_c} \lambda_k \frac{\partial}{\partial r_i} g_k(r),$$  

(1.81)

and the Lagrange multipliers can be determined by enforcing the new position to satisfy the constraint condition (Equation 1.80)

$$
\begin{align*}
\mathbf{r}_i(t + \Delta t) &= \mathbf{r}^{\text{unconstr}}_i(t + \Delta t) + \frac{1}{2} m_i^{-1} \Delta t^2 f^{\text{constr}}_i(t) \\
g(\mathbf{r}(t + \Delta t)) &= 0
\end{align*}
$$

(1.82)

The positions $\mathbf{r}^{\text{unconstr}}_i(t + \Delta t)$ are the unconstrained positions, obtained by a (leap-frog) integration step only using the unconstrained forces

$$f^{\text{unconstr}}_i = - \frac{\partial}{\partial r_i} V'(r).$$

(1.83)

A well known example using this approach to introduce distance constraints (and bond-angle constraints) into molecular dynamics simulations is the SHAKE algorithm\textsuperscript{54}. A variant of this algorithm to constrain dihedral angles is given in Section 6.6.2.

If two macro-states of a system can be distinguished by a different value of $\langle Q \rangle_s$ where the index $s$ indicates averaging over all micro-states belonging to the corresponding macro-state, $\langle Q \rangle$ is called an order parameter of the transition from one state to the other. An order parameter is distinct from a reaction (or transition) coordinate, as it does not describe a detailed (microscopic) path from one state to the other, but merely characterises a configuration as belonging to a certain state or a transition. The difference in the Helmholtz free energy $A$ (in the canonical ensemble)

$$A_s(N,V,T) = -k_B T \ln(Z_s(N,V,T))$$

(1.84)

or the Gibbs free enthalpy $G$ (in the isothermic-isobaric ensemble)

$$G_s(N,V,T) = A_s(N,V,T) + pV = -k_B T \ln(Z_s(N,p,T))$$

(1.85)

with $Z_s$ the partition function of state $s$, between two macro-states $a$ and $b$ is related to the equilibrium constant $K_{ab} = N_b/N_a$ with $N_a$ and $N_b$ the number of conformations belonging to states $a$ or $b$, respectively, as follows

$$\Delta A_{ab} = -1/k_B T \ln(K_{ab}).$$

(1.86)

In this manner, a simulation that sufficiently samples both macro-states $A$ and $B$ and also the transition between those two states can be used directly to determine the relative free energy of these states (using free energy either for the Helmholtz free energy or the Gibbs free enthalpy, depending on which ensemble is generated in the simulation). This direct-counting approach has been used in an extended-Lagrangian scheme called $\lambda$-dynamics\textsuperscript{55} to calculate series of free energy differences.
For many cases, the occurrence of a transition between the two states is too infrequent to apply direct counting and the transition has to be considered as a rare event. Sampling of rare events occurring at time scales much longer than the duration of a numerical simulation is only possible if the transition is forced during the simulation time. This implies some form of interference with the dynamics of the system. Several approaches are available\textsuperscript{14}. Specifying both the initial and final state in terms of some order parameter and sampling the trajectories connecting them, so-called transition path sampling\textsuperscript{56}, may be least intrusive. Recently, an improvement of the method for rate constant calculation called transition interface sampling has been introduced\textsuperscript{57} and reviewed\textsuperscript{58}.

The free energy difference of two states may also be obtained by calculating the work necessary to get from state \(a\) to state \(b\)\textsuperscript{59}

\[
\Delta A_{ab} = W_{ab} = \int_a^b \left\langle \frac{\partial}{\partial r} \mathcal{H}(r, p) \right\rangle_r dr. \tag{1.87}
\]

As the Helmholtz free energy as well as the Gibbs free enthalpy are state functions and therefore path independent, one is free to choose any order parameter of the transition to define a path from \(a\) to \(b\)

\[
W_{ab} = \int_{q_a}^{q_b} \left\langle \frac{\partial}{\partial q} \mathcal{H}(r, p) \right\rangle_q dq, \tag{1.88}
\]

where \(\langle \ldots \rangle_q\) means the ensemble (or time) average of a system with a constrained microscopic observable \(q\). If the system contains other constraints than the constrained order parameter defining the transition path, Equation 1.88 contains additional correction factors\textsuperscript{60}.

If the two states are not distinguished by an order parameter but rather by a different Hamiltonian a similar approach may be taken. First, a transition path leading from Hamiltonian \(\mathcal{H}_a\) of state \(a\) to the Hamiltonian \(\mathcal{H}_b\) of state \(b\) is defined

\[
\mathcal{H}(r, p, \lambda) = (1 - \lambda)\mathcal{H}_a(r, p) + \lambda\mathcal{H}_b(r, p), \tag{1.89}
\]

using a coupling parameter \(\lambda\)\textsuperscript{61,62}. Setting \(\lambda = 0\) corresponds to state \(A\), while a \(\lambda\) value of 1 represents state \(B\). At intermediate \(\lambda\)-values the Hamiltonian is a linear combination of the two states. Using either Equation 1.84 or Equation 1.85 yields the thermodynamic integration formula

\[
\Delta A_{ab} = \int_a^b \left\langle \frac{\partial \mathcal{H}(r, p, \lambda)}{\partial \lambda} \right\rangle_\lambda d\lambda. \tag{1.90}
\]

The free energy difference can now be calculated numerically by changing \(\lambda\) during one simulation (or many short simulations\textsuperscript{63-65}) or by numerically integrating over the ensemble averages obtained at different \(\lambda\)-values using many simulations.

A general improvement of the searching properties of a molecular dynamics simulation may be achieved by the local elevation technique\textsuperscript{66}. The technique of conformational flooding is based on the same idea\textsuperscript{67} and the so-called meta-dynamics method is basically another form
of local elevation\textsuperscript{68,69}. The local elevation method improves sampling by adding a memory-dependent repulsive potential energy term to the interaction function, which penalises re-visiting of conformations encountered before. To determine whether a conformation has been visited before, it is (coarsely) represented in terms of dihedral angles, but the method may be formulated using other degrees of freedom as classification criterion as well. Due to combinatorial explosion, enhancing the sampling is limited to a relatively small set of dihedral angles. In Chapter 7 a new method to use the local elevation technique in biomolecular structure refinement is introduced.

Sampling may be substantially enhanced, without introducing any (non-dynamical) bias, using the replica-exchange method\textsuperscript{70-72} (or a closely related method developed independently\textsuperscript{73}). The replica-exchange method is also known as multiple Markov chain method\textsuperscript{74} and parallel tempering\textsuperscript{75}. Details of the method can be found in recent reviews\textsuperscript{76,77} and its implementation into GROMOS is given in Section 2.4.2, “Replica-exchange simulation”. In the method, a number of non-interacting replicas of the original system are propagated at different temperatures (or using different Hamiltonians), either by Monte Carlo or molecular dynamics simulation methods\textsuperscript{78}. Every few steps, pairs of replica may be interchanged, with an exchange probability related to their difference in potential energy. The method has been widely applied, including applications to the polypeptide folding problem\textsuperscript{79-86}. However, as the number of degrees of freedom of the system increases, the required number of replicas also greatly increases, which reduces the efficiency of the method substantially when large systems are simulated. A combination of the replica-exchange method with generalized ensemble algorithms, which might alleviate this problem, has been reviewed recently\textsuperscript{87}.

1.4 Deterministic chaos

In the past decades a large amount of publications and books were written on the topic of chaos and nonlinear dynamics. For this introduction, some were considered especially helpful\textsuperscript{88-90}.

Chaos originally meant the infinite empty space which existed before all things, which during the Roman era changed to mean the original crude shapeless mass into which the Architect of the world would introduce order and harmony. Modern usage of the term chaos denotes a state of disorder and irregularity. Classical mechanics are deterministic. One would think that deterministic equations of motion lead to a regular behaviour of the system in time. But already at the beginning of the century, Poincaré discovered that certain mechanical systems governed by Hamilton’s equations of motion could display chaotic motion\textsuperscript{91}.

If we knew exactly the laws of nature and the situation of the universe at the initial moment, we could predict exactly the situation of that same universe at a succeeding moment. But even if it were the case that the natural laws had no longer any secret for us, we could still only know the initial situation approximately. If that enabled us to predict the succeeding situation with the same approximation, that is
all we require, and we should say that the phenomenon had been predicted, that it is governed by laws. But it is not always so; it may happen that small differences in the initial conditions produce very great ones in the final phenomena. A small error in the former will produce an enormous error in the latter. Prediction becomes impossible, and we have the fortuitous phenomenon.

— Henri Poincaré

Still, over half a century passed until with the discovery by Edward Norton Lorenz (born 1917) the phenomenon became more than a mere curiosity. Lorenz found that even a simple set of three coupled, first order, nonlinear differential equations can lead to completely chaotic trajectories, hence deterministic chaos. Lorenz called this sensitive dependence on the initial conditions the butterfly effect, because the outcome of the equations, which, in a crude way, describe the flow of air in the earth’s atmosphere, could be changed by a butterfly flapping its wings.

Introduction of the notion of chaos has impact on many areas of physics. An interesting example is the connection between chaotic dynamics and statistical mechanics. The second law of thermodynamics ($\frac{d}{dt} S \geq 0$ for isolated systems) leads to irreversibility. This poses a significant problem to classical physics, as irreversibility implies a preferred direction of time (for a macroscopic system). But the laws of classical dynamics do not change when the direction of time is reversed. Ludwig Eduard Boltzmann (1844–1906) proposed a statistical model which accurately predicts macroscopic values of thermodynamic quantities and explained irreversibility on a simple model system, albeit relying on a (statistical) assumption (a similar but mathematically simpler treatment is provided by Baker). The assumption necessary in Boltzmann’s treatment may be explained physically and for simple cases even proven through the introduction of chaotic dynamics. A simple way to proceed from simple statistical assumptions to the laws of thermodynamics was proposed in 1928 by Gilbert Lewis and Joseph Mayer.

Boltzmann’s assumption states that in a dilute gas, where only binary collisions occur (to simplify the mathematics, not strictly necessary), there should be molecular chaos. Molecular chaos (also referred to as “Stosszahlansatz”) is defined as a loss of correlation between positions and velocities. One can think of molecular chaos represented by a macro-state corresponding to an average over nonessential or unmeasurable micro-states. To introduce molecular chaos into the simple model of a dilute gas, he assumed that after collisions particles loose all memory of their previous velocities. Therefore, velocity and position become uncorrelated with each other, and knowledge only of the distribution of velocities remains. This assumption also contains an implicit reference to the sequence of events (collisions) and in that manner direction of time, and with that it explains irreversibility in Boltzmann’s statistical model.

With the discovery of chaotic behaviour it was possible to show for a hard sphere gas that the trajectories of particles with only small differences of initial positions diverge exponentially with time (on average), making a prediction of the trajectories impossible. Therefore, chaotic
dynamics may provide a mechanism for justifying statistical mechanics.

With the link between chaotic dynamics and statistical mechanics established, the predictive powers of molecular dynamics simulations stand to question. As long as a simulation of a true macro-state (containing a large number of micro-states) is impossible, a thoughtful selection of initial states seems mandatory to obtain trajectories that represent a physical situation, because this selection will determine the subspace of phase space that is sampled. Often, it can be shown that a specific property of interest converges to the same value over a simulation, even if different initial conditions are used.

On the other hand, chaotic behaviour may lead to mixing (correlation functions decay to zero in the infinite time limit), which in turn assures quasi-ergodicity (phase space averages can be replaced by time averages).

Chaotic dynamics do not necessarily contradict Pierre-Simon de Laplace’s (1749 – 1827) view of causal determinism99,100 (and translated101). Problematic is rather whether it is possible to determine all properties of any given state perfectly, which is necessary for a long-term prediction, or just approximately. This was expressed in 1860 by James C. Maxwell (1831 – 1879)102.

_It is a metaphysical doctrine that from the same antecedents follow the same consequents. No one can deny this. But it is not much use in a world like this, in which the same antecedents never again concur, and nothing ever happens twice._

_The physical axiom which has a somewhat similar aspect is “that from like antecedents follow like consequents”. But here we have passed from sameness to likeness, from absolute accuracy to a more or less rough approximation. There are certain classes of phenomena, in which a small error in the data only introduces a small error in the result, the course of events in these cases is stable. There are other classes of phenomena which are more complicated, and in which cases instability may occur, the number of such cases increasing, in an extremely rapid manner, as the number of variables increases._

— James Clark Maxwell
1.5 Outline

In Chapter 2 an overview of the next version of GROMOS: GROMOS05 will be given, with special emphasis on MD++, the new MD-engine written in C++. MD++ has been written with special attention to modularity and simplicity. This eased the implementation of the algorithms proposed in subsequent chapters tremendously.

The next chapter will provide a review of the various searching and sampling methods available today, providing an introduction to the sampling techniques used in Chapter 4, 5, 6 and 7. Specifically, in Chapter 4 the residual influence on entropy of elevated temperature in the history of any replica in the replica-exchange method will be investigated, and in Chapter 5 a combination of coarse-graining and fine-graining using the replica-exchange method to improve sampling will be presented.

Sampling restrained to a transition path in between two states, but without restraining the end-states themselves is the focus of Chapter 6. Restraints have also been used to keep properties in a simulation close to experimental values as is done in biomolecular structure refinement. Restraining the simulation but also enhancing sampling as long as the restraints are not fulfilled yet seems to be important to satisfy $^3J$-value restraints. A combination of local-elevation with $^3J$-value restraining accomplishing this will be shown in Chapter 7.

Finally, in the next two chapters a fast, but approximate way of using constraints, with added flexibility, instead of restraints is introduced and the free energy differences of a change from water to methanol using either a (hard) constrained or a flexible constrained or a fully flexible (similar to restrained) model for the bonds between the atoms are compared.

At last, the thesis will be concluded by a short outlook.

1.6 Bibliography


1.6. Bibliography


1.6. Bibliography


Chapter 2

The GROMOS software for biomolecular simulation: GROMOS05

2.1 Summary

The latest version of the Groningen Molecular Simulation program package, GROMOS05 is presented. It has been developed for the dynamical modelling of (bio)molecules using the methods of molecular dynamics, stochastic dynamics, and energy minimisation. An overview of GROMOS05 is given, highlighting features not present in the last major release, GROMOS96. The organisation of the program package is outlined and the included analysis package GROMOS++ is described. Finally, some applications illustrating the various available functionalities are presented.
2.2 Introduction

Starting with GROMOS80, the GROMOS program package has been developed over the past 25 years to facilitate research efforts in the field of biomolecular simulation in a university environment. The GROMOS software was and is meant for use in a scientific environment, which may be characterised by a continuously changing flow of users, who either wish to investigate and implement new simulation algorithms or intend to carry out applications of simulation in a variety of fields, ranging from polymers, glasses and liquid crystals to crystals and solutions of biomolecules (proteins, nucleic acids, saccharides and lipids). To this purpose GROMOS has been developed based on the following principles: (i) transparency of the code, making modifications easy, (ii) modular architecture, so that only parts of it need be modified for the implementation of new functionalities designed by users, (iii) independence of the simulation code and the force field, and (iv) independence of the simulation code and the computer hardware.

Figure 2.1: Distribution of GROMOS licences.

The major releases of the GROMOS software are GROMOS87\textsuperscript{1,2} developed at the University
of Groningen, GROMOS96\textsuperscript{3,4} developed at ETH Zürich, and now GROMOS05. GROMOS has found widespread use (hundreds of licences in over 57 countries on all continents except Antarctica, see Figure (2.1)), triggered by the fact that it has been designed for ease of extendability and that the complete source code is made available to research establishments for a nominal fee\textsuperscript{5}. The program code has been further developed in the group for computational chemistry at ETH Zürich (Switzerland) throughout the recent years, leading now to a new major release, GROMOS05. The enhancements were governed by the following criteria: (1) interest of our research group\textsuperscript{6}, (2) ease of use, (3) extendability, (4) demonstrated usefulness or efficiency of new methods, (5) well-defined and correct formulae and algorithms, and (6) computational efficiency. The second criterion led to a complete rewrite of the setup and analysis tools, now contained in the GROMOS++ setup and analysis subpackage, written in C++. The third criterion led to a rewrite in C++ of the MD engine, the part that carries out molecular dynamics (MD) or stochastic dynamics (SD) simulations as well as energy minimisations (EM), into a new program called MD++. In parallel, the original FORTRAN version of the MD engine, PROMD, was further developed to introduce many new features (some of which are not yet available in MD++). On the long term (beyond GROMOS05), MD++ will entirely replace PROMD.

In the next sections the main features of GROMOS05 are described. In Section 2.3 new functionalities with respect to GROMOS96 are highlighted and in Section 2.4 follows the algorithmic description of selected new functionalities. In Section 2.5, the organisation of the code is discussed and an overview of the programs present in the GROMOS++ analysis subpackage is provided. In Section 2.6, examples of applications are reported for some of the newer features. Section 2.7 provides a summary and conclusions.

In this thesis the focus is on the features and implementation of MD++, more details regarding PROMD are provided elsewhere\textsuperscript{7}.

2.3 Overview of functionalities

Here, the main features of the two MD engines available in GROMOS05, PROMD and MD++ are described. These two programs share most of the basic functionalities, but still differ in a number of aspects. The FORTRAN MD engine (PROMD) retains all features of the GROMOS96 release and adds a number of new functionalities. The C++ MD engine (MD++) contains most of the GROMOS96 features (except four-dimensional and path integral simulations), a subset of the new functionalities recently introduced into PROMD (since the GROMOS96 release), and some new features of its own.

A non exhaustive list of the features included is:

- Molecular dynamics (MD), stochastic dynamics (SD) simulation and energy minimisation (EM; steepest descent or conjugate gradient);
• Periodic boundary conditions (vacuum, rectangular, truncated octahedral or triclinic computational box; possibility of performing multiple-unit-cell simulations);

• Temperature control (constraining, weak coupling, Nosé-Hoover or Nosé-Hoover chain; possible coupling of different subsets of degrees of freedom to separate temperature baths);

• Pressure control (weak coupling or Andersen-Parrinello-Rahman: isotropic, partially anisotropic and fully anisotropic coordinate scaling; atom-based or group-based pressure definition);

• Long-range electrostatic interactions: straight cutoff truncation, truncation with Poisson-Boltzmann reaction field (RF) correction and lattice-sum (LS; PROMD only; details provided elsewhere) methods, including Ewald summation and particle-particle-particle-mesh (P³M);

• Charge-group based or atom-based cutoff for the non-bonded interactions;

• Grid based pairlist construction;

• Non-physical interactions: atom-position, atom-distance, dihedral-angle, NOE and J-value restraints as well as atom-position and atom-distance constraints (SHAKE, M-SHAKE, LINCS), hidden (distance and dihedral angle) restraints (see Chapter 6) and adaptive (J-value) restraints (see Chapter 7);

• Enhanced sampling: local elevation MD, replica exchange MD (REMD), multigraining (see Chapter 5) and umbrella sampling;

• Calculation of free energy changes based on the coupling parameter (\( \lambda \)) approach using thermodynamic integration, slow-growth or one-step perturbation, possibly including soft-core nonbonded interactions;

• Path integral simulation (PROMD only);

• MPI and OMP parallelisation;

A number of the new features introduced in GROMOS05 are discussed in Section 2.4.2. Pre-existing features have been described in details elsewhere\(^3\). The functionalities of the pre- and post-processing programs contained in GROMOS++ are discussed in Section 2.5.3. A complete description of the available features will be included in the new GROMOS manuals.
2.4 Algorithms

2.4.1 MD algorithm

The complete MD algorithm based on the leap-frog scheme as implemented in GROMOS is the following. Given initial atomic positions and velocities, which satisfy any given geometrical constraints:

1. Save positions (reset atomic coordinates into the reference computational box in case of periodic boundary conditions) and velocities for later analysis.

2. Remove centre of mass motion (if required).

3. Calculate (unconstrained) energies, forces and virial contribution from the potential energy function (using the nearest image convention in case of periodic boundary conditions). Save these.

4. Enforce any given position constraints by resetting the forces and velocities of positionally constrained atoms to zero.

5. Update the velocities using the leap-frog scheme.

6. Apply temperature coupling (constraining, weak coupling, Nosé-Hoover or Nosé-Hoover chain) by scaling the atom velocities.

7. Update the positions using the leap-frog scheme.

8. Enforce distance constraints (using SHAKE, M-SHAKE or LINCS) both for positions and velocities, and calculate the corresponding forces and virial contribution. Save these.

9. Calculate the kinetic energy and temperature (possibly on the basis of separate subsets of degrees of freedom).

10. Calculate the pressure (atom-based or group-based pressure definition).

11. Apply pressure scaling (weak coupling or Parrinello-Rahman) by scaling atomic positions (isotropic, partially anisotropic or fully anisotropic scaling).

12. Update the coupling parameter $\lambda$ for (free energy) simulations involving $\lambda$ changes (slow growth).

13. Calculate total energies, averages and fluctuations. Save these.

This sequence is repeated for the required number of simulation steps.
2.4.2 New features

Spatial boundary conditions

Spatial boundary conditions are defined by the shape, size and orientation of the simulated system, and the nature of the boundary to its surroundings. The GROMOS05 implementation (both PROMD and MD++) admits four types of boundary conditions: (i) vacuum boundary conditions; (ii) periodic boundary conditions based on a rectangular box; (iii) periodic boundary conditions based on a truncated-octahedral box; (iv) periodic boundary conditions based on a triclinic box.

In the three latter cases, the system is confined to a (reference) computational box that is surrounded by an infinite number of periodic copies of itself.

When periodic boundary conditions are applied, the shape, size and orientation of the computational box must be defined. For rectangular and triclinic periodic boundary conditions, this is done by specifying the three edge vectors \( \mathbf{a}, \mathbf{b}, \text{ and } \mathbf{c} \) (defining a right-handed coordinate system) of the reference computational box. For a truncated-octahedral box, these vectors correspond instead to the edges of the cube based on which the truncated octahedron is constructed. In practice, the three vectors are specified by their lengths \( a, b, \text{ and } c \), the box angles \( \alpha \) (between \( \mathbf{a} \text{ and } \mathbf{b} \)), \( \beta \) (between \( \mathbf{a} \text{ and } \mathbf{c} \)) and \( \gamma \) (between \( \mathbf{b} \text{ and } \mathbf{c} \)) they define among each other (all in the range \([0; \pi]\)), and the three Euler rotation angles \( \phi, \theta \text{ and } \psi \) (the two former ones in the range \([-\pi; \pi]\), the latter one in the range \([-\pi/2; \pi/2]\)) characterising the orientation of the box relative to the reference right-handed Cartesian coordinate system \((\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)\). To define the Euler angles, the three edge vectors are used to define a box-linked right-handed Cartesian coordinate system \((\mathbf{e}_x', \mathbf{e}_y', \mathbf{e}_z')\) in the following way: (i) the \( x'\)-axis is chosen along and in the direction of \( \mathbf{a} \); (ii) the \( y'\)-axis is chosen orthogonal to \( \mathbf{a} \) in the plane defined by \( \mathbf{a} \text{ and } \mathbf{b} \), and oriented in the direction of \( \mathbf{b} \); (iii) the \( z'\)-axis is chosen orthogonal to both \( \mathbf{a} \) and \( \mathbf{b} \), and oriented in the direction of \( \mathbf{c} \).

The reference coordinate system can be rotated onto the box-linked coordinate system by the following series of rotations: (i) a rotation by an angle \( \phi \) around the \( z'\)-axis; (ii) a rotation by an angle \( \theta \) around the new \( y'\)-axis; (iii) a rotation by an angle \( \psi \) around the new \( x'\)-axis. The angles \( \phi, \theta \text{ and } \psi \) thus represent the three Euler rotation angles in a \( zyx \) or yaw-pitch-roll convention.

The use of a rectangular or truncated-octahedral box requires \( \alpha = \beta = \gamma = \pi/2 \) and is restricted to non-rotated boxes with \( \phi = \theta = \psi = 0 \). The use of a truncated-octahedral box also requires \( a = b = c \). In the case of vacuum boundary conditions, the system is non-periodic, and \( \mathbf{a}, \mathbf{b}, \text{ and } \mathbf{c} \) need not be specified.

Based on a general triclinic box in an arbitrary orientation, the position of an atom may be specified through coordinates \( \mathbf{r} = (x, y, z) \) within the reference Cartesian coordinate system \((\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)\), or through oblique fractional coordinates \( \tau = (u, v, w) \) with reference to the box-edge vectors. The two types of coordinates are related by

\[
\mathbf{r} = \mathbf{L} \tau ,
\]

where the matrix \( \mathbf{L} \) contains the components of \( \mathbf{a}, \mathbf{b}, \text{ and } \mathbf{c} \) in the reference Cartesian coordinate system.
system as its columns. The box volume is

\[ V = |\mathbf{L}|. \] (2.2)

This matrix can be decomposed as

\[
\mathbf{L} = \begin{pmatrix}
 a_x & b_x & c_x \\
 a_y & b_y & c_y \\
 a_z & b_z & c_z
\end{pmatrix} = \mathbf{RS},
\] (2.3)

where the orthogonal transformation matrix \( \mathbf{R} \) (rotation between reference and box-linked Cartesian coordinate systems) is given by

\[
\mathbf{R} = \begin{pmatrix}
 \cos \theta \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\
 \cos \theta \sin \phi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & -\cos \psi \sin \theta \sin \phi + \sin \psi \cos \phi \\
 -\sin \theta & \sin \psi \cos \theta & \cos \psi \cos \theta
\end{pmatrix},
\] (2.4)

and the transformation matrix \( \mathbf{S} \) (between box-linked Cartesian coordinates and oblique fractional coordinates) is given by

\[
\mathbf{S} = \begin{pmatrix}
 a & b \cos \gamma & c \cos \beta \\
 0 & b \sin \gamma & c \sin \beta \cos \delta \\
 0 & 0 & c \sin \beta \sin \delta
\end{pmatrix},
\] (2.5)

with

\[
\cos \delta = \frac{\cos \alpha - \cos \beta \cos \gamma}{\sin \beta \sin \gamma}, \quad \delta \in [0; \pi[.
\] (2.6)

As shown by Bekker\(^8\), a simulation performed in a truncated-octahedral box can equivalently be performed in a special type of triclinic box, by applying an appropriate coordinate transformation. A possible choice for the edges \( a_t, b_t \) and \( c_t \) of the transformed triclinic box is

\[
a_t = a, \quad b_t = (1/2)(a + b + c) \quad \text{and} \quad c_t = (1/2)(-a - b + c).
\] (2.7)

The corresponding box-edge lengths, box angles and Euler angles are \( a_t = a, \ b_t = c_t = (\sqrt{3}/2)a, \ \alpha_t = \cos(-1/3) \approx 109.5^\circ, \ \beta_t = \cos(-1/\sqrt{3}) \approx 125.3^\circ, \ \gamma_t = \cos(1/\sqrt{3}) \approx 54.8^\circ, \ \phi_t = \theta_t = 0, \) and \( \psi_t = 45^\circ. \) The mapping of atomic coordinates within a truncated-octahedral box to atomic coordinates within the transformed triclinic box is performed by applying shifts along the \( a_t, b_t \) and \( c_t \) vectors. This formalism is applied for the generalisation of grid-based pairlist algorithms (see Pairlist construction) and lattice-sum electrostatics (details elsewhere\(^7\)) to truncated-octahedral boxes. Because the truncated-octahedral case can always be mapped to the triclinic case, subsequent sections will only discuss the case of the triclinic box.
Multiple-unit-cell simulations

Within the GROMOS05 implementation (both PROMD and MD++), it is possible to simulate a periodic computational box (rectangular or triclinic only) consisting of multiple periodic copies of a smaller unit cell (referred to here as subcells). This option may be useful when trying to simulate a single unit cell of a crystal that is too small to allow for the application of a reasonably large cutoff value. The number of subcell boundaries along the three box-edge vectors \( \mathbf{a} \), \( \mathbf{b} \) and \( \mathbf{c} \) are \( M_a \), \( M_b \) and \( M_c \), so that the total number of subcells is \( M = M_a \cdot M_b \cdot M_c \).

In MD++ only a single subcell is simulated. Just for the non-bonded interaction calculation the subcell is multiplied to construct the full reference cell. Energies, forces and virial contributions need only be calculated for atoms inside this reference subcell, but these atoms are interacting with all other atoms in the full cell. Because of that, less (non-bonded and covalent) interactions than in the full reference cell simulation have to be calculated and the positions (and velocities) in the subcells are always exactly periodic.

Note that the removal of the center of mass motion (see Section 2.4.2, "Rigid-body motion"), whenever required, is applied to charge groups and solvent molecules gathered in the individual subcells.

Rigid-body motion

The laws of classical mechanics lead to two conserved quantities (besides the total energy): (i) the linear momentum \( \mathbf{p}_{\text{sys}} \) of the system, and (ii) the angular momentum \( \mathbf{L}_{\text{sys}} \) of the system around its center of mass. In simulations under periodic boundary conditions, the two quantities refer to the infinite periodic system. However, in this case, if the linear momentum \( \mathbf{p}_{\text{box}} \) of the computational box is also conserved, the corresponding angular momentum \( \mathbf{L}_{\text{box}} \) is not (because correlated rotational motions in two adjacent boxes exert friction on each other, leading to an exchange of kinetic energy with the other degrees of freedom of the system). Furthermore, the quantity \( \mathbf{L}_{\text{sys}} \) must vanish (because overall uniform rotation of the infinite periodic system would lead to non-periodic centrifugal forces). When SD is applied instead of MD, the presence of random and frictional forces couple the system (or box) linear and angular momenta with the other degrees of freedom of the system, so that these quantities are no longer conserved. The inclusion of special (unphysical) forces, such as atom-position restraining or constraining forces on a subset of atoms in the system, may also lead to non-conservation of these quantities. The above observations are summarised in Table 2.1.

The physical properties of a molecular system are independent of \( \mathbf{p}_{\text{sys}} \) (or \( \mathbf{p}_{\text{box}} \)). However, for MD simulations under vacuum boundary conditions, they depend on \( \mathbf{L}_{\text{sys}} \), because the rotation of the system leads to centrifugal forces. For these reasons, in the GROMOS05 implementation, the constraint \( \mathbf{p}_{\text{sys}} = 0 \) (or \( \mathbf{p}_{\text{box}} = 0 \)) may be imposed at each timestep throughout any simulation. In addition, the constraint \( \mathbf{L}_{\text{sys}} = \mathbf{L}^\circ_{\text{sys}} \), where \( \mathbf{L}^\circ_{\text{sys}} \) is a user-specified reference value, may be imposed throughout any MD simulation (PROMD only), or \( \mathbf{L}_{\text{sys}} \) may be constrained to
Table 2.1: Properties of momenta associated with rigid-body motions in MD or SD simulations under vacuum or periodic boundary conditions. The quantities considered are: $p_{\text{sys}}$ and $p_{\text{box}}$ (linear momentum of the overall system and the computational box), $L_{\text{sys}}$ and $L_{\text{box}}$ (angular momentum of the overall system and the computational box), and $N_r$ (number of uncoupled degrees of freedom associated with rigid-body motions).

<table>
<thead>
<tr>
<th>method</th>
<th>boundary</th>
<th>$p_{\text{sys}}$</th>
<th>$p_{\text{box}}$</th>
<th>$L_{\text{sys}}$</th>
<th>$L_{\text{box}}$</th>
<th>$N_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD</td>
<td>vacuum</td>
<td>conserved</td>
<td>conserved</td>
<td>conserved</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>MD</td>
<td>periodic</td>
<td>infinite</td>
<td>conserved</td>
<td>zero</td>
<td>coupled</td>
<td>3</td>
</tr>
<tr>
<td>SD</td>
<td>vacuum</td>
<td>coupled</td>
<td>coupled</td>
<td>coupled</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>periodic</td>
<td>infinite</td>
<td>coupled</td>
<td>zero</td>
<td>coupled</td>
<td>0</td>
</tr>
</tbody>
</table>

0 (MD++), under vacuum boundary conditions. These two constraints will in particular prevent the progressive accumulation of kinetic energy into the uncoupled degrees of freedom due to applying a thermostat by velocity scaling (see Section 2.4.2, "Instantaneous temperature and pressure") and numerical errors, giving rise to the well-known (and quite unpleasant) “flying ice cube problem”9-11. As an alternative, in MD++ roto-translational constraints12 may be applied to the solute molecule(s) during the simulation.

**Instantaneous temperature and pressure**

The instantaneous observables $T$ and $\mathbf{P}$, the time averages of which determine the system macroscopic temperature $T$ and pressure tensor $\mathbf{P}$, are not uniquely defined9,13-16. Acceptable alternative definitions differ by any quantity with a vanishing equilibrium average. Note, however, that the corresponding equilibrium fluctuations depend on the specific definition chosen for the instantaneous observable.

In the GROMOS05 implementation (both PROMD and MD++), the instantaneous temperature $T$ is defined using the (atom-based) internal kinetic energy of the system, as9

$$T = \frac{2}{k_B N_{df}} \mathcal{K},$$  \hspace{1cm} (2.8)

where $k_B$ is Boltzmann’s constant, $N_{df}$ the number of internal (unconstrained) degrees of freedom of the system and $\mathcal{K}$ its instantaneous internal kinetic energy. The word “internal” is used here to exclude possible contributions from the degrees of freedom that are “external”, i.e. uncoupled from the system in terms of kinetic energy exchange17. In MD simulations, these are the degrees of freedom associated with the system (or box) rigid-body translation and, under vacuum boundary conditions, system rigid-body rotation (see Section 2.4.2, “Rigid-body motion”). The number of internal degrees of freedom is thus calculated as three times the total number $N$ of
atoms in the system, minus the number $N_c$ of geometrical constraints, minus the number $N_r$ of external degrees of freedom (2.1), i.e.

$$N_{df} = 3N - N_c - N_r.$$  \hfill (2.9)

The instantaneous internal kinetic energy is defined as

$$\mathcal{K} = \frac{1}{2} \sum_{i=1}^{N} m_i \mathbf{v}_i^2,$$  \hfill (2.10)

where the internal (also called peculiar) velocities $\mathbf{v}_i$ are obtained from the real atomic velocities $\mathbf{v}_i^0$ by excluding any component along the external degrees of freedom (it is assumed that the velocities $\mathbf{v}_i^0$ are already exempt of any component along possible geometrical constraints). Due to the constraints imposed in the GROMOS05 implementation on the system total linear and angular momenta (see Section 2.4.2, “Rigid-body motion”), the internal velocities only differ from the real ones when MD is applied under vacuum boundary conditions with a non-zero angular momentum. In this case, one has

$$\mathbf{v}_i = \mathbf{v}_i^0 - \mathbf{I}^{-1}_{CM} \mathbf{r}_i^0 \times (\mathbf{r}_i^0 - \mathbf{r}_CM^0),$$  \hfill (2.11)

where $\mathbf{r}_{CM}$ is the (constant) coordinate vector of the system center of mass, $\mathbf{I}^0_{sys}$ the (constant) system angular momentum about the CM, and $\mathbf{I}^0_{CM}$ is the (configuration-dependent) inertia tensor of the system relative to the CM. The latter quantity is defined as

$$\mathbf{I}^0_{CM}(\mathbf{r}^0) = \sum_{i=1}^{N} m_i (\mathbf{r}_i^0 - \mathbf{r}_{CM}^0) \otimes (\mathbf{r}_i^0 - \mathbf{r}_{CM}^0),$$  \hfill (2.12)

where $\mathbf{a} \otimes \mathbf{b}$ denotes the tensor with elements $\mu, \nu$ equal to $a_\mu b_\nu$.

In the GROMOS05 implementation (both PROMD and MD++), the instantaneous pressure tensor $\mathbf{P}$ is related to the group-based virial and group-based internal kinetic energy tensor of the system. The word “group-based” refers to a pressure definition excluding virial and kinetic-energy contributions within user-specified groups of (covalently-linked) atoms\textsuperscript{15,16}. These groups will be referred to as virial groups. Single atoms can be used as virial groups, in which case an atom-based pressure definition is recovered. The average pressure is not affected by the specific choice of groups, but the pressure fluctuations are. In practice, atom grouping is used to reduce these fluctuations. The pressure is only calculated for systems under periodic boundary conditions. Note also that the contribution of special (non-physical) forces (e.g. atom-position or atom-distance restraining) to the pressure is not included.

The instantaneous atom-based pressure tensor is computed as

$$\mathbf{P}^* = \frac{2}{N} (\mathcal{K}^* - \mathcal{W}^*),$$  \hfill (2.13)
2.4. Algorithms

where
\[
\mathbf{K} = \frac{1}{2} \sum_{i=1}^{N} m_i \mathbf{r}_i \otimes \mathbf{r}_i ,
\]
and
\[
\mathbf{W}_{\mu \nu} = \frac{1}{2} \sum_{\lambda} \frac{\partial U}{\partial L_{\mu \lambda}} I_{\nu \lambda}.
\]
are the instantaneous atom-based internal kinetic energy and virial tensors, \( U \) and \( U \) being the instantaneous volume and total potential energy of the system, \( L \) the matrix defined by Equation (2.3) and \( \mathbf{r}_i \) the internal velocities introduced above. The corresponding isotropic (scalar) quantities are related to the tensor quantities through
\[
\mathbf{K} = \text{Tr}[\mathbf{K}], \quad \mathbf{W} = \text{Tr}[\mathbf{W}] \quad \text{and} \quad \mathbf{P} = \frac{1}{3} \text{Tr}[\mathbf{P}],
\]
where \( \text{Tr} \) returns the trace of a matrix, \( \mathbf{K} \) is equivalent to \( \mathbf{K} \) in Equation (2.10) and \( \mathbf{W} \) is defined as
\[
\mathbf{W} = \frac{3 \nu \partial U}{\partial 

It is possible to show that\(^{18,19}\): (i) the contribution to the atom-based virial tensor of a potential energy term that solely depends on the scalar products or determinants defined by a set of interatomic vectors is symmetric; (ii) the contribution to the atom-based virial tensor of a potential energy term that solely depends on the angles defined by a set of vectors is (in addition) traceless. The first observation implies that all covalent (bond-stretching or constraint, bond-angle bending, proper and improper dihedral-angle) and pairwise non-bonded force-field terms lead to a symmetric contribution to the atom-based virial. The second observation implies that covalent bond-angle bending as well as proper and improper dihedral-angle (but not bond-stretching or constraint and pairwise non-bonded) terms lead to a traceless contribution to the atom-based virial (\textit{i.e.} no contribution to the scalar atom-based pressure). However, these results are generally not valid for the corresponding group-based tensor (see below).

In the special case of a pairwise-additive interaction term \( U_p \) depending on minimum-image interatomic distances and without explicit dependence on the box dimensions (bond-stretching or constraint and pairwise non-bonded terms; but not reciprocal-space lattice-sum interactions\(^{15,16}\)), Equation (2.15) leads to a virial contribution
\[
\mathbf{W}_p = -\frac{1}{2} \sum_{i}^{N} \sum_{j>i}^{N} \mathbf{F}_{p,ij} \otimes \mathbf{r}_{ij},
\]
where \( \mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j \) is the vector connecting \( j \) to \( i \), \( \mathbf{r}_{ij} \) the corresponding minimum-image vector and \( \mathbf{F}_{p,ij} \) the pairwise force exerted by atom \( j \) on atom \( i \). This equation is easily generalised to interaction terms involving more than two atoms (bond-angle bending, proper and improper
dihedral- angle terms; see Section 1.1). The atom-based virial contribution of all covalent (including bond constraints) and non-bonded (excluding reciprocal-space lattice-sum interactions) terms is calculated using Equation (2.18) or one of its generalisations.

The GROMOS05 implementation (both PROMD and MD++) includes the possibility of using a group-based pressure definition (corresponding to any arbitrary repartition of subsets of covalently-linked atoms into virial groups), instead of the atom-based one. In this case, the intra-group contribution to the kinetic energy as well as the contribution of intra-group forces to the virial are removed from the pressure definition (which affects the fluctuations of this quantity, but not its average value). As shown elsewhere\textsuperscript{15,16} (the equations reported therein should be altered by halving the virial and replacing $r_{ij}$ by $-r_{ij}$ to match the present conventions), the group-based virial tensor can be calculated from the corresponding atom-based tensor by adding a simple correction term which depends on the overall atomic forces and on the submolecule definitions. More precisely, the group-based virial tensor is given by

$$W = W^* + \frac{1}{2} \sum_{i \alpha} F_{i \alpha} \otimes d_{i \alpha},$$  \hspace{1cm} (2.19)

where $i \alpha$ denotes atom $\alpha$ in the virial group $I$, $F_{i \alpha}$ the overall force on atom $i \alpha$, and $d_{i \alpha}$ the coordinate vector of atom $i \alpha$ relative to the center of mass of the gathered virial group $I$ containing this atom. The “gathered” representation of the virial group is generated by following the atoms as they drift throughout the periodic system. The group-based pressure tensor is then calculated as

$$P = \frac{2}{\nu} (\mathcal{K} - W),$$  \hspace{1cm} (2.20)

where $\mathcal{K}$ is the group-based internal kinetic energy tensor, defined as

$$\mathcal{K} = \frac{1}{2} \sum_{l=1}^{N_s} \left( \sum_{\alpha=1}^{N_\alpha(l)} m_i \right)^{-1} \left( \sum_{\alpha=1}^{N_\alpha(l)} m_i \hat{r}_i \right) \otimes \left( \sum_{\alpha=1}^{N_\alpha(l)} m_i \hat{r}_i \right),$$  \hspace{1cm} (2.21)

where $N_s$ is the number of submolecules and $N_\alpha(I)$ the number of atoms in submolecule $I$.

Although the atom-based pressure tensor $P^*$ is typically symmetric, this is generally not the case for the group-based pressure tensor $P$ (although the anti-symmetric contribution to this tensor should vanish upon time averaging). When applying a barostat algorithm (see Barostat algorithms), the antisymmetric component of $P$ should induce an overall rotation of the computational box (which would alter the box angular momentum), while the symmetric component results in a deformation of the box (which conserves the box angular momentum). In practice, the overall rotation of the box is rather a nuisance, and is avoided by symmetrising the tensor $(P \rightarrow (1/2)(P + tP))$ prior to application of the barostat algorithm\textsuperscript{20}, where the “t” pre-superscript indicates the transpose of the matrix.
Thermostat algorithms

MD simulation relies on integrating the classical (Newtonian) equations of motion for a molecular system and thus, samples a microcanonical (constant-energy) ensemble by default. However, for compatibility with experiment, it is often desirable to sample configurations from a canonical (constant-temperature) ensemble instead. A modification of the basic MD scheme with the purpose of maintaining the temperature constant (on average) is called a thermostat algorithm. Note that in contrast, SD automatically generates a canonical ensemble, at a temperature determined by the balance between the magnitudes of the random and frictional forces.

In the GROMOS05 implementation (both PROMD and MD++), four different thermostat algorithms are available: (i) temperature constraining (Woodcock thermostat); (ii) temperature relaxation by weak-coupling (Berendsen thermostat); (iii) temperature relaxation by an extended-system method (Nosé-Hoover thermostat); (iv) temperature relaxation by the Nosé-Hoover-chain thermostat. In all cases, the instantaneous temperature $T$ is calculated as described in Section 2.4.2, "Instantaneous temperature and pressure", and relaxed towards a temperature $T_0$ associated with the heat bath to which the system is coupled. The three latter algorithms also involve the specification of the characteristic time $\tau$ for this relaxation. All the above thermostat algorithms rely on a scaling of the atomic velocities after each integration timestep. This scaling should only operate on the internal velocities, excluding any component along the external degrees of freedom (Section 2.4.2, "Instantaneous temperature and pressure"). Due to the constraints imposed in the GROMOS05 implementation on the system total linear and angular momenta (Section 2.4.2, "Rigid-body motion"), the internal velocities only differ from the real ones when MD is applied under vacuum boundary conditions. In this case, the velocity scaling is applied on the internal velocities $\mathbf{\dot{r}_i}$ and the real velocities $\mathbf{\dot{r}_i}^0$ can be recovered through the inverse of Equation (2.11), namely

$$\mathbf{\dot{r}_i}^0 = \mathbf{\dot{r}_i} + L^{-1}_{CM}(\mathbf{r}^0) \cdot L_{sys}^0 \times (\mathbf{\dot{r}_i} - \mathbf{\dot{r}_CM}). \quad (2.22)$$

When simulating molecular systems involving distinct sets of degrees of freedom with either (i) very different characteristic frequencies or (ii) very different heating (or cooling) rates caused by algorithmic noise (e.g. electrostatic cutoff, application of atom-distance constraints), the joint coupling of all degrees of freedom to a single thermostat may lead to different effective temperatures for the different sets of degrees of freedom (due to a too slow exchange of kinetic energy). A typical example is the so-called "hot solvent - cold solute problem" in simulations of macromolecules. Because the solvent is often more significantly affected by algorithmic noise (heating due to the use of an electrostatic cutoff), the coupling of the whole system to a single thermostat may cause the average solute temperature to be significantly lower than the average solvent temperature. In the GROMOS05 implementation (both PROMD and MD++), this problem may be alleviated by separately coupling different subsets of degrees of freedom (e.g. solute, counter-ions, co-solvent and solvent) to different independent thermostats.
The prototype of most isothermal equations of motion is
\[ \dot{r}_i(t) = m_i^{-1} F_i(t) - \gamma(t) \dot{r}_i(t). \] (2.23)

The function \( \gamma(t) \) controls the heat exchange between the system and the heat bath. A negative value indicates that heat flows to the system, while a positive value indicates a heat flow in the opposite direction. Practical implementations of Equation (2.23) rely on the stepwise integration of Newton’s second law (Equation (2.23) with \( \gamma(t) = 0 \)), altered by the scaling of the atomic velocities after each iteration step. In the context of the leap-frog integrator used in GROMOS05, this can be written as
\[ \dot{r}_i(t + \frac{\Delta t}{2}) = \lambda(t; \Delta t) \dot{r}_i(t + \frac{\Delta t}{2}) = \lambda(t; \Delta t) \left[ \dot{r}_i(t - \frac{\Delta t}{2}) + m_i^{-1} F_i(t) \Delta t \right], \] (2.24)

where \( \lambda(t; \Delta t) \) is a time- and timestep-dependent velocity scaling factor. Imposing the constraint \( \lambda(t; 0) = 1 \), one recovers Equation (2.23) in the limit of an infinitesimal timestep \( \Delta t \), with
\[ \gamma(t) = -\frac{\partial \lambda(t; \Delta t)}{\partial (\Delta t)} \bigg|_{\Delta t=0}. \] (2.25)

The Woodcock thermostat (also known as temperature constraining thermostat; see also the Hoover-Evans thermostat) aims at fixing the instantaneous temperature \( T \) exactly at the reference heat-bath value \( T_0 \), without allowing for any fluctuations. In this case, the quantity \( \lambda(t; \Delta t) \) in Equation (2.24) is found by imposing \( T(t + \frac{\Delta t}{2}) = \frac{g}{N_{df}} T_0 \), leading to
\[ \lambda(t; \Delta t) = \left[ \frac{g}{N_{df}} T(t + \frac{\Delta t}{2}) \right]^{1/2}. \] (2.26)

where \( T(t + \frac{\Delta t}{2}) \) is the temperature evaluated from the velocities \( \dot{r}_i(t + \frac{\Delta t}{2}) \) in Equation (2.24). The corresponding quantity \( \gamma(t) \) in Equation (2.23) is given by
\[ \gamma(t) = (gk_B T_0)^{-1} \sum_{i=1}^{N} \dot{r}_i(t) \cdot F_i(t). \] (2.27)

Although \( g = N_{df} \) seems to be the obvious choice, it turns out that \( g = N_{df} - 1 \) is the appropriate one for the algorithm to generate a canonical distribution of configurations (though obviously not of momenta) at temperature \( T_0 \). The reason is that constraining the temperature effectively removes one degree of freedom from the system. Note, however, that the absence of kinetic energy fluctuations may lead to inaccurate dynamics, especially in the context of the microscopic systems typically considered in simulations.

The Berendsen thermostat (also known as weak-coupling thermostat) aims at relaxing the instantaneous temperature \( T \) to the reference heat-bath value \( T_0 \) based on a first-order (weak-coupling) scheme with a characteristic time \( \tau_B \), i.e. as
\[ \dot{T}(t) = \tau_B^{-1} [T_0 - T(t)]. \] (2.28)
2.4. Algorithms

In this case, the quantity \( \lambda(t; \Delta t) \) in Equation (2.24) is found by imposing
\[
T(t + \Delta t) = T(t - \Delta t) + \tau_B^{-1} \Delta t \frac{g}{N_{df}} \left[ T_o - T(t - \Delta t) \right],
\]
where in principle \( g = N_{df} \), leading to
\[
\lambda(t; \Delta t) = \left\{ \frac{T(t - \Delta t)}{T'(t + \Delta t)} + \tau_B^{-1} \Delta t \frac{g}{N_{df}} \frac{T_o - T(t - \Delta t)}{T'(t + \Delta t)} \right\}^{1/2}
\approx \left\{ 1 + \tau_B^{-1} \Delta t \left[ \frac{g}{N_{df}} \frac{T_o}{T'(t + \Delta t)} - 1 \right] \right\}^{1/2}. \tag{2.29}
\]

In GROMOS05, the algorithm is implemented following the second (approximate) expression. For either of the two expressions, the corresponding quantity \( \gamma(t) \) in Equation (2.23) is given by
\[
\gamma(t) = \frac{1}{2} \tau_B^{-1} \left[ \frac{g}{N_{df}} \frac{T_o}{T(t)} - 1 \right]. \tag{2.30}
\]

In practice, \( \tau_B \) is used as an empirical parameter to adjust the strength of the coupling to the heat-bath. In the limit \( \tau_B = \Delta t \), the Berendsen algorithm is equivalent to the Woodcock algorithm (and thus generates a canonical distribution of configurations, but not of momenta). In the limit \( \tau_B \to \infty \), the thermostat becomes inactive and the Newton equation of motion is recovered (which samples a microcanonical ensemble). However, except in the former limit (and only for the configurational part), the ensemble generated by the Berendsen equations of motion is not a canonical ensemble.

The Nosé-Hoover thermostat\(^2\) aims at relaxing the instantaneous temperature \( T \) to the reference heat-bath value \( T_o \) based on an extended-system approach with a characteristic time \( \tau_{NH} \). In the original Nosé algorithm\(^3\), the real system is extended by addition of an artificial \((N_{df} + 1)^{th}\) (positive) dynamical variable \( s \) (associated with a "mass" \( Q > 0 \) as well as a velocity \( \dot{s} \)), that plays the role of a time-scaling parameter. Through an appropriate choice for the extended-system Lagrangian, a microcanonical MD trajectory in the extended-system can be mapped onto a canonical trajectory in the real system. However, the Nosé thermostat leads to sampling of the real-system trajectory at uneven time intervals, which is quite impractical. This inconvenience is alleviated by rewriting the equations of motion in terms of the real-system variables, as was later shown simultaneously by Nosé\(^3\) and Hoover\(^4\). In the Nosé-Hoover algorithm, the quantity \( \gamma(t) \) in Equation (2.23) is not uniquely determined by the instantaneous microstate of the system, but is a dynamical variable which derivative is determined by this instantaneous microstate through
\[
\dot{\gamma} = -\tau_{NH}^{-2} \frac{T_o}{T} \left( \frac{g}{N_{df}} \frac{T_o}{T} - 1 \right), \tag{2.31}
\]
where the effective coupling time \( \tau_{NH} \) is related to the (less intuitive) effective mass \( Q \) in the Nosé thermostat through
\[
\tau_{NH} = (N_{df} k_B T_o)^{-1/2} Q^{1/2}. \tag{2.32}
\]
When $\gamma$ is negative, heat flows from the heat bath into the system due to Equation (2.23). When the system temperature increases above $T_0$, the time derivative of $\gamma$ becomes positive in Equation (2.31) and the heat flow is progressively reduced (feedback mechanism). Conversely, when $\gamma$ is positive, heat is removed from the system until the system temperature decreases below $T_0$ and the heat transfer is slowed down. In practice, Equation (2.31) is discretised (based on the simulation timestep $\Delta t$) and integrated simultaneously with the equations of motion for the atomic coordinates and velocities based on the leap-frog scheme.

It can be proven\(^9,24\) that the Nosé-Hoover equations of motion sample a canonical ensemble (in both coordinates and momenta) provided that $g = N_{HF}$ and that $\tau_{NH}$ is finite, this irrespective of the actual value of $\tau_{NH}$ and of the initial conditions for the atomic velocities and for the $\gamma$ variable.

In practice $\tau_{NH}$ is used as an empirical parameter to adjust the strength of the coupling to the heat-bath. Too large values of $\tau_{NH}$ (loose coupling) may cause a poor temperature control (the limiting case of the Nosé-Hoover thermostat with $\tau_{NH} \to \infty$ and $\gamma(0) = 0$ is MD, which generates a microcanonical ensemble). On the other hand, too small values (tight coupling) may cause high-frequency temperature oscillations leading to the same effect.

The Nosé-Hoover-chain thermostat\(^25\) aims at relaxing the instantaneous temperature $T$ to the reference heat-bath value $T_0$ based on a chain of successive thermostat variables. In this case the single thermostat variable $\gamma$ of the Nosé-Hoover scheme is replaced by a chain of variables applying a thermostat to each other in sequence. This algorithm has been introduced to alleviate the two main drawbacks of the Nosé-Hoover algorithm: (i) the presence of temperature oscillations, and (ii) the non-ergodicity of the sampling for small or stiff systems, or systems at low temperatures\(^24,31-36\). The GROMOS05 implementation follows the formalism described in the original article\(^25\).

**Barostat algorithms**

For compatibility with experiment, it is often desirable to sample configurations from the isothermal-isobaric ensemble (constant temperature and pressure). Thermostat algorithms have been described above (Section 2.4.2, "Thermostat algorithms"). A modification of the basic MD scheme with the purpose of maintaining the pressure constant (on average) is called a barostat algorithm.

The use of a barostat is only applicable to simulations under periodic boundary conditions. In the GROMOS05 implementation (both PROMD and MD++), the various options for the variations of the box parameters (and the associated scaling of atomic coordinates) involved in the use of a barostat are: (i) no variations of the box parameters; (ii) isotropic scaling, i.e. identical relative variations of the box-edge lengths only; (iii) partially anisotropic scaling, i.e. independent relative variations of the box-edge lengths only; (iv) fully anisotropic scaling, i.e. independent variations of all box parameters (box-edge lengths, box angles and Euler angles). For a truncated-
2.4. Algorithms

octahedral box, only the first two options are allowed. For a rectangular box, only the first three options are allowed. For a triclinic box, all options are allowed. In the latter case, variations in the box shape are accompanied by variations in the box Euler angles, so as to guarantee that the barostat does not introduce a rigid-body rotational component to the box orientation. Note, however, that the location of the box center of mass is affected by any type of coordinate scaling.

Two different barostat algorithms will be available: (i) pressure relaxation by weak-coupling (Berendsen barostat); (ii) pressure relaxation by extended-system method (Andersen-Parrinello-Rahman barostat; implementation in progress).

Pairlist construction

The evaluation of the non-bonded interactions in GROMOS relies on the application of the twin-range method. The GROMOS05 implementation (both PROMD and MD++) of this approach includes an increased amount of flexibility, and relies on the definition of: (i) a short-range pairlist distance $R_p$; (ii) a corresponding cutoff distance $\tilde{R}_p \leq R_p$ (optional); (iii) a lower-bound for the intermediate-range pairlist distance $R_l$; (iv) a corresponding cutoff distance $\tilde{R}_l \geq R_l$ (optional); (v) an upper-bound for the intermediate-range pairlist distance $R_l$; (vi) a corresponding cutoff distance $\tilde{R}_l \leq R_l$ (optional); (vii) a short-range pairlist update frequency $N_s$; (viii) an intermediate-range pairlist update frequency $N_l$. Short-range interactions are computed every timestep based on a short-range pairlist containing pairs in the distance range $[0; R_p]$, or a filtered subset of this list corresponding to pairs currently (i.e. at the given timestep) in the distance range $[0; \tilde{R}_p]$. The short-range pairlist is reevaluated every $N_s$ timesteps. It can be generated either on the basis of distances between charge-groups (groups of covalently linked atoms defined in the system topology) or of distances between individual atoms. In the former case, the filtering (based on the distance $\tilde{R}_p$) may be based either on distances between charge-groups or on distances between atoms. In the latter case, only atom-based filtering is possible. Intermediate-range interactions are computed every $N_l$ timesteps based on all pairs in the distance range $[R_l; R_l]$, or a filtered subset of these pairs in the distance range $[\tilde{R}_l; \tilde{R}_l]$ at the time of the evaluation of these interactions. Only an atom-based filtering is possible here, and it is only meaningful when the initial set of pairs is generated on the basis of distances between charge-groups. The energy, forces and virial contributions associated with intermediate-range interactions are assumed constant between two updates (i.e. during $N_l$ steps).

The evaluated interaction includes Lennard-Jones and electrostatic components. The latter component may include a reaction-field contribution (Section 2.4.2, “Reaction-field electrostatics”) or the real-space contribution to a lattice-sum method (details elsewhere). Note that the real-space contribution to a lattice-sum method may only be computed within the short-range contribution to the interaction.

The pairlist construction may be performed in four different ways: (i) using the standard double-loop algorithm included in the GROMOS96 program; (ii) using a different, more efficient approach based on a fast-converging saddle point search; (iii) using a combination of the standard double-loop algorithm and the fast-converging saddle point search; (iv) using a completely different approach based on a different method for generating the pairlist.

The evaluated interaction includes Lennard-Jones and electrostatic components. The latter component may include a reaction-field contribution (Section 2.4.2, “Reaction-field electrostatics”) or the real-space contribution to a lattice-sum method (details elsewhere). Note that the real-space contribution to a lattice-sum method may only be computed within the short-range contribution to the interaction.

The pairlist construction may be performed in four different ways: (i) using the standard double-loop algorithm included in the GROMOS96 program; (ii) using a different, more efficient approach based on a fast-converging saddle point search; (iii) using a combination of the standard double-loop algorithm and the fast-converging saddle point search; (iv) using a completely different approach based on a different method for generating the pairlist.
possibility of an atom-based cutoff and of filtering); (ii) using an optimised version which improves processor cache usage (details elsewhere); (iii) using a grid-based pairlist algorithm introduced recently (PROMD only); (iv) using a slight variation of the above grid-based algorithm which permits easier parallelisation and avoids periodicity corrections during the interaction evaluation (MD++ only).

**Grid-based pairlist construction.** PROMD includes a recently introduced grid-based pairlist algorithm that permits the fast construction of cutoff-based non-bonded pairlists in molecular simulations under periodic boundary conditions based on an arbitrary box shape (rectangular, truncated-octahedral or triclinic). The key features of this algorithm are: (i) the use of a one-dimensional mask array (to determine which grid cells contain interacting atoms) that incorporates the effect of periodicity, and (ii) the grouping of adjacent interacting cells of the mask array into stripes, which permits the handling of empty cells with a very low computational overhead. Testing of the algorithm on water systems of different sizes (containing about 2000 to 11000 molecules) has shown that the method: (i) is about an order of magnitude more efficient compared to a standard (double-loop) algorithm, (ii) achieves quasi-linear scaling in the number of atoms, (iii) is weakly sensitive in terms of efficiency to the chosen number of grid cells.

MD++ includes a slightly modified version of this grid-based pairlist algorithm extending on ideas similar to those of a published pairlist algorithm. Grid-based pairlist algorithms are more efficient than a standard double loop pairlist generation because only a reduced set of neighbouring atoms is considered to determine whether the atom-pair is within the cutoff distance. Within this scheme of a grid-based pairlist algorithm, additional efficiency advantages may be realised. First, nearest image calculations for atom-pairs can be avoided. Second, the virial can be calculated from the total force on the atoms instead of computing the contribution of each pair. And third, the atoms may be ordered in memory, therefore allowing linear access during the pairlist generation and force calculations. In the first step, the system gets extended on all sides by the cutoff-distance, where the additional atom or charge-group positions are obtained by simple shifts of the original positions by the lattice vectors. At the same time the grid-cell index of each charge-group (or atom) is calculated. As second step, the atom positions are reordered according to their (or the corresponding charge-group's) grid-cell index and the starting index of each grid-cell is stored. Using this information plus a one-dimensional interaction mask containing the offsets of all grid-cells within cutoff distance of the current grid-cell, a short-range pairlist of (contiguous) ranges of atoms is generated, where (for a charge-group based cutoff criterion) complete charge-group pairs are excluded if they contain excluded atom pairs. The long-range interactions (energies and forces) are calculated directly during the pairlist generation. Finally, the short-range interactions are calculated and the missing interactions between atom pairs in previously excluded charge-groups are added.

The virial tensor is calculated from a sum of contributions of all atom pairs, which leads to the following expression compatible with a grid-based pairlist generation (comparable to a
published virial tensor calculation

\[ W = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=-13}^{13} w(i, j, k) \mathbf{r}_{ij} \otimes \mathbf{f}_{ij}, \]  

(2.33)

where

\[ \mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j = \mathbf{r}_i - \mathbf{r}_j - \mathbf{t}_k \]  

(2.34)

and \( \mathbf{t}_k \) is the vector to shift \( \mathbf{r}_j \) from the central computational \((k = 0)\ box to one of its 26 direct neighbours \((k = -13, \ldots, 0, \ldots, 13; \text{including also the central box itself with } t_k = 0)\). To include each interaction only once, the weight factor \( w(i, j, k) \) is defined using the grid-cell index \( g(i) \) of atom \( i \) (and the grid-cell index \( g(j, k) \) of atom \( j \) shifted by \( \mathbf{t}_k \))

\[ w(i, j, k) = \begin{cases} 
1 & \text{if } g(\mathbf{r}_i) < g(\mathbf{r}_{jk}) \text{ or } g(\mathbf{r}_i) = g(\mathbf{r}_{jk}) \text{ and } i < j \\
 & \text{and } \mathbf{r}_{i,jk} < R_p \\
0 & \text{otherwise}
\end{cases} \]  

(2.35)

Using Equation 2.34 the expression for the virial tensor (Equation 2.35) can be rewritten to

\[ W = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=-13}^{13} w(i, j, k) \mathbf{r}_i \otimes \mathbf{f}_{ij} + \\
\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=-13}^{13} w(i, j, k) \mathbf{r}_j \otimes \mathbf{f}_{ji} + \\
\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=-13}^{13} w(i, j, k) \mathbf{t}_k \otimes \mathbf{f}_{jk} \]  

= \( A + B + C \).  

(2.36)

Exchanging \( i \) and \( j \) in the second term \((B)\) leads to

\[ B = \sum_{j=1}^{N} \sum_{i=1}^{N} \sum_{k=-13}^{13} w(j, i, k) \mathbf{r}_i \otimes \mathbf{f}_{ik}, \]  

(2.37)

and reordering the two summations gives

\[ B = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=-13}^{13} w(j, i, k) \mathbf{r}_i \otimes \mathbf{f}_{ik}. \]  

(2.38)

And therefore adding up \( A \) and \( B \) yields

\[ A + B = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=-13}^{13} w(i, j, k) \mathbf{r}_i \otimes \mathbf{f}_{jk} + w(j, i, k) \mathbf{r}_i \otimes \mathbf{f}_{ik} \]  

(2.39)

\[ = \sum_{i=1}^{N} \mathbf{r}_i \otimes \sum_{j=1}^{N} \sum_{k=-13}^{13} w(i, j, k) \mathbf{f}_{jk} + w(j, i, k) \mathbf{f}_{ik}, \]  

(2.40)
where the latter part is identical to the total force on atom $i$

$$f_i = \sum_{j=-13}^{13} \sum_{k=-13}^{13} w(i, j_k) f_{ijk} + w(j, i_k) f_{ikj}. \tag{2.41}$$

Using that, Equation 2.39 simply is

$$A + B = \sum_{i=1}^{N} r_i \otimes f_i. \tag{2.42}$$

The remaining third term from Equation 2.36 term $C$ (the periodicity correction)

$$C = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=-13}^{13} w(i, j_k) t_k \otimes f_{jk}. \tag{2.43}$$

is easily calculated from the forces on the atoms in the previously extended areas around the central computational box

$$f_{jk} = \sum_{i=1}^{N} w(i, j_k) f_{ji}. \tag{2.44}$$

as given by

$$C = \sum_{k=-13}^{13} t_k \otimes \sum_{j=1}^{N} f_{jk}. \tag{2.45}$$

In summary, the contribution to the virial tensor due to the nonbonded interactions may be calculated outside of the inner loop (the loop over all atom-pairs within the cutoff) through Equations 2.42 and 2.45,

$$\mathcal{W} = \sum_{i=1}^{N} r_i \otimes f_i + \sum_{k=-13}^{13} t_k \otimes \sum_{j=1}^{N} f_{jk}. \tag{2.46}$$

In the framework of the grid-based pairlist algorithm using an extended system, the total forces $f_i$ on atom $i$ and the partial forces $f_{jk}$ on the atom shifted by $t_k$ are readily available. Because of the ordered memory layout of the atoms and because of storing ranges of interacting atoms in the pairlist (where the exclusions have been removed already), automatic vectorization of the code is possible.

A comparison of the overall efficiency of GROMOS96 (standard double loop pairlist construction and an optimized version that improves processor cache usage) with MD++ is given in Table 2.2; timings are given for complete simulations including pairlist construction and force calculation. The MD++ version was not particularly optimized.
Table 2.2: Efficiency comparison of GROMOS96 (FORTRAN (PROMD), standard pairlist algorithm (STD) and pairlist algorithm with optimised cache usage (OPT)) and MD++ (C++; standard pairlist algorithm (STD) and grid-based pairlist algorithm (GRID)). As test systems liquid alkane (23328 solute atoms, 9.4 x 9.4 x 9.4 nm cubic box), a membrane (6656 solute atoms, 7383 solvent (SPC water) atoms, 6.2 nm x 6.2 nm x 6.9 nm rectangular box) and a protein (2445 solute atoms, 47472 solvent (SPC water) atoms, 7.9 nm x 7.9 nm x 8.3 nm rectangular box) were used. All calculations were done on a dual processor AMD Athlon MP 248 PC (2000 MHz processor frequency, 512 KB cache, 2 GB RAM). The efficiency was measured running on a single processor and running in parallel on both processors, 250 simulation steps for the alkane and membrane system, 100 steps for the protein. All numbers are in seconds. For the grid based MD++ simulations, time spent in the nonbonded interaction calculation is indicated in brackets.

Reaction-field electrostatics

When cutoff truncation is applied to the Coulombic interactions within a molecular system, the mean effect of the omitted electrostatic interactions beyond the (long-range) cutoff distance $R_l$ (Section 2.4.2, “Pairlist construction”) may be approximately reintroduced through a so-called reaction-field correction term. This approximation relies on assuming that the medium beyond the cutoff sphere of each particle (i.e. beyond a specified distance $R_{RF}$, typically set equal to $R_l$) is a linearised-Poisson-Boltzmann continuum characterised by a relative dielectric permittivity $\varepsilon$ and an inverse Debye screening length $\kappa$. In the present context, these two parameters may be combined into an effective permittivity

$$\varepsilon_{RF} = \left[ 1 + \frac{(\kappa R_{RF})^2}{2(\kappa R_{RF} + 1)} \right] \varepsilon.$$  \hspace{1cm} (2.47)

In the GROMOS05 implementation (both PROMD and MD++), the corresponding overall...
electrostatic energy (Coulomb plus reaction-field term) is then written\(^\text{(2.48)}\):

\[
q_{el}^{CB+RF} = \frac{1}{4\pi\varepsilon_0} \left\{ \sum_i \sum_{j>i,j\notin\text{excl}(i)} q_i q_j \left( \frac{1}{r_{ij}^2} + \frac{2(\varepsilon_{RF} - 1)}{2\varepsilon_{RF} + 1} \frac{1}{2R_{RF}^3} - \frac{3\varepsilon_{RF}}{2\varepsilon_{RF} + 1} \frac{1}{R_{RF}} \right) \right. \\
+ \sum_i \sum_{j>i,j\notin\text{excl}(i)} q_i q_j \left( \frac{2(\varepsilon_{RF} - 1)}{2\varepsilon_{RF} + 1} \frac{1}{2R_{RF}^3} - \frac{3\varepsilon_{RF}}{2\varepsilon_{RF} + 1} \frac{1}{R_{RF}} \right) \\
\left. - \frac{1}{2\varepsilon_{RF} + 1} \frac{R_{RF}}{R_{RF}} \left\{ \sum_i q_i^2 - \varepsilon_{RF}^{-1} \left( \sum_i q_i \right)^2 \right\} \right\},
\]

where \(\varepsilon_0\) is the permittivity of vacuum, \(\vec{r}_{ij}\) is the minimum-image vector corresponding to \(r_{ij} = r_i - r_j\), and \(\text{excl}(i)\) denotes the exclusion list of atom \(i\) (including its first and second covalent neighbours; the distance between any two excluded atoms is assumed to be smaller than \(R_i\)). Note that current simulation programs (e.g. GROMOS96\(^3\) and GROMACS\(^55\)) typically restrict the implementation of the reaction-field method to the first term in Equation (2.48). The second term is explicitly included here because excluded neighbours should only be exempted from the direct (Coulombic) interaction and not from the solvent-mediated (reaction-field) interaction\(^54\).

The form of the third term has been chosen for consistency in the context of small molecules (compared to the cutoff radius and box size). For such a small molecule (or ion) gathered by periodicity around its center, \(\vec{r}_{ij}\) can be replaced by \(r_{ij}\) and the cutoff truncation involved in the first summation of Equation (2.48) can be omitted. In this case, it can be shown\(^54\) that the reaction-field contribution to \(q_{el}^{CB+RF}\) for a neutral molecule matches the correct Onsager expression for the solvation of a dipolar molecule in a spherical cavity\(^56\) (for \(\kappa = 0\)). For a monoatomic ion, the last term can also be shown\(^54\) to represent a (first-order) correction to the error in solvation free energy caused by the use of effective (non-Coulombic) interactions. Intuitively, this last term may be interpreted as the reversible work required to individually charge the atoms when they are at infinite separation. This contribution only affects the energy of the system, but does not induce atomic forces. However, it may be essential to include it in free-energy calculations involving alterations of the atomic partial charges and comparisons between different media (\(\varepsilon\)).

**Replica-exchange simulation**

To obtain canonical distributions for complex molecular systems, efficient sampling of the configurational space is necessary. Finding the global minimum on the typically rough potential energy landscape of a peptide or protein is likewise difficult. In recent years, the replica-exchange method\(^57\)-\(^62\) (also known as parallel tempering\(^62\)) has received much attention. A number of non-interacting replicas are simulated simultaneously at different conditions (e.g. different temperatures). After a given simulation time, an exchange between two replicas is attempted, followed by another (individual) simulation period. The method has been applied to biomolecular
2.4. Algorithms

systems\textsuperscript{63–66}, using Monte-Carlo techniques and molecular dynamics (REMD) to propagate the individual replicas. The probability of each state \( x = (r, p) \) in the canonical ensemble at temperature \( T \) is proportional to the weight factor

\[
W(x) = \exp(-\beta H(r, p)) . \tag{2.49}
\]

where \( H \) is the Hamiltonian and \( \beta = 1/k_B T \), \( k_B \) being Boltzmann’s constant. The weight factor for the global state \( X \) determined by the states of the \( M \) replicas is the product of the single weights, \textit{i.e.}

\[
W_{\text{REM}}(X) = \exp\left(-\sum_{i=1}^{M} \beta_i H(r_i, p_i)\right) . \tag{2.50}
\]

After a fixed number of MD integration steps, a Monte-Carlo (MC) exchange between two replicas is attempted (changing from state \( X \) to state \( X' \)). In order to sample canonical ensembles at each temperature, the detailed balance condition on the transition probability \( W(X \to X') \)

\[
W_{\text{REM}}(X)w(X \to X') = W_{\text{REM}}(X')w(X' \to X) \tag{2.51}
\]

has to be fulfilled. This can be satisfied, for instance, by the usual Metropolis criterion

\[
p(X \to X') = \frac{w(X \to X')}{w(X' \to X)} = \left\{ \begin{array}{ll}
1 & \text{for } \Delta \leq 0, \\
\exp(-\Delta) & \text{for } \Delta > 0
\end{array} \right., \tag{2.52}
\]

with

\[
\Delta = (\beta_i - \beta_j)(U(r_j) - U(r_i)). \tag{2.53}
\]

where \( U(r) \) is the potential energy associated with the configuration \( r \). If the exchange was successful, the momenta of the exchanged replicas are scaled to correspond to their new temperatures.

An extension of the replica-exchange method to sample the isothermal-isobaric ensemble has been suggested\textsuperscript{67}. In this case, an additional term incorporating the pressure and volume change appears in the exchange probability

\[
\Delta = (\beta_i - \beta_j)(U(r_j) - U(r_i)) + (\beta_i P_i - \beta_j P_j)(V_j - V_i). \tag{2.54}
\]

Unfortunately, the application of the method to explicit solvent simulations, though successful for small systems, is rather difficult\textsuperscript{68–74}. Since the exchange probability between two states decreases with increasing system size, explicit-solvent simulation requires many more states (replicas) separated by small temperature differences. This problem may be alleviated by not exchanging a thermodynamic property like the temperature between the replicas, but rather altering specific interactions\textsuperscript{75–77}. Then, the replicas are distinguished by their Hamiltonians

\[
H_i(r_i, p_i) = K(p_i) + U_i(r_i) \tag{2.55}
\]
using for each replica a different potential energy function $U_i$. In MD++ the different Hamiltonians $H_i$ are defined using a coupling parameter $\lambda$ and a perturbation topology. The replica with $\lambda_i = 0$ corresponds to state A (normal topology) in a perturbation simulation, the replica at $\lambda_i = 1$ to state B (fully perturbed state; with, e.g. scaled down non-bonded or bonded interaction terms. The other replicas are distributed in between these two ($0 < \lambda_i < 1$).

Inserting the individual $H_i$ into Equation (2.50), leads to

$$W_{REM}(X) = \exp \left( - \sum_{i=1}^{M} \beta_i H_i(\mathbf{r}_i, \mathbf{p}_i) \right),$$

and, using the detailed balance criterion (Equation (2.51)) to

$$p(X \to X') = \frac{w(X \to X')}{w(X' \to X)} = \begin{cases} 1 & \text{for } \Delta \leq 0, \\ \exp(-\Delta) & \text{for } \Delta > 0 \end{cases},$$

with

$$\Delta = \beta_i(U_i(\mathbf{r}_j) - U_i(\mathbf{r}_i)) - \beta_j(U_j(\mathbf{r}_j) - U_j(\mathbf{r}_i)).$$

where the potential energy of the two configurations $\mathbf{r}_i$ and $\mathbf{r}_j$ needs to be evaluated twice, with Hamiltonians $H_i$ and $H_j$.

The replica exchange method was implemented in MD++ based on sockets and TCP as communication protocol. A server distributes the short MD runs corresponding to the different replicas to a (dynamical) number of clients. After the given number of simulation steps has been carried out, the client reports back to the server the final (potential) energies (evaluated using the Hamiltonian $H_i$ and the Hamiltonian $H_j$ (if different), the server then calculates the switching probability $p(i \to j)$, draws a random number and, if the switch is successful, exchanges the states. As soon as a client is free, the next replica gets assigned to it.

Replicas can differ in the temperature and in the coupling parameter $\lambda$. A replica-exchange state consists of replicas for all possible $\lambda_i$ values at each temperature $T_i$ ($M = N_\lambda \cdot N_T$ replicas). The Monte-Carlo exchange attempt is alternated between exchanges of (neighbouring) $\lambda$'s and temperatures $T$.

**Coarse-grained simulation**

Most molecular simulations are making use of atom-level (AL) models. This limits the time scale of such simulations for solvated macromolecules to the nanosecond range. Longer time scales can be reached by treating molecules or molecular fragments as single particles or beads, whose motion is simulated using a simple force field describing inter-bead interactions. When the energy function of such a coarse-grained (CG) model is chosen to be smooth and short-ranged, the efficiency of CG simulations can be orders of magnitude ($10^3 - 10^5$) higher than the corresponding AL simulations, be it at the expense of the loss of atomic detail and some accuracy.$^{78-82}$
A recently proposed CG model for liquid simulations has the same functional form as the GROMOS force field, except for the use of a switching function for the non-bonded Lennard-Jones and electrostatic interactions at distances just below the cutoff distance. This CG model was implemented into GROMOS05 (MD++ only), however with a slightly different switching function, because the GROMACS one appeared to be discontinuous and led to non-conservation of energy in MD simulation.

In the absence of switching, the non-bonded interaction energy between particles \(i\) and \(j\) can be written as \(V(r_{ij}) = \sum_{\alpha=1,6,12} V_\alpha(r_{ij}) = \sum_{\alpha=1,6,12} c_\alpha \Phi_\alpha(r_{ij})\)

\[ V(r_{ij}) = \sum_{\alpha=1,6,12} V_\alpha(r_{ij}) = \sum_{\alpha=1,6,12} c_\alpha \Phi_\alpha(r_{ij}) \quad (2.59) \]

with \(r = |r|\)

\[ \Phi_\alpha(r) = r^{-\alpha} \quad (2.60) \]

where standard notations for atomic charges \(q_i\) and van der Waals interaction parameters \(C_6\) and \(C_{12}\) have been used. In the CG model all non-bonded interactions are smoothly switched to zero over the range \([R_{sw}, R_c]\), where \(R_{sw}\) denotes the start of the switching and \(R_c\) the cut-off radius. In this model one has \(R_c = 1.2 \text{ nm}\) and \(R_{sw} = 0 \text{ nm}\) for the Coulomb interaction and \(R_{sw} = 0.9 \text{ nm}\) for the van der Waals interactions. The non-bonded interaction energy function including switching reads for the three terms in Equation (2.59)

\[ \Phi^s_\alpha(r) = \begin{cases} \Phi_\alpha(r) & r \leq R_{sw} \\ \Phi_\alpha(r) + S_\alpha(r) & R_{sw} \leq r \leq R_c \\ 0 & r \geq R_c \end{cases} \quad (2.62) \]

Requiring that the functions \(S_\alpha(r)\) switch the energy, the force and the derivative of the force smoothly (without discontinuities) to zero at \(r = R_c\), yields the conditions

\[ S_\alpha(R_{sw}) = S'_\alpha(R_{sw}) = S''_\alpha(R_{sw}) = 0 \quad (2.63) \]

and

\[ \Phi^s_\alpha(R_c) = \Phi'_\alpha(R_c) = \Phi''_\alpha(R_c) = 0 . \quad (2.64) \]

The conditions of Equation (2.63) are satisfied by a fourth-degree polynomial

\[ S_\alpha(r) = -\frac{1}{3} A (r - R_{sw})^3 - \frac{1}{4} B (r - R_{sw})^4 - C . \quad (2.65) \]
The conditions Equation (2.64) determine the constants

\[ A = \frac{\alpha[(\alpha + 1)R_{sw} - (\alpha + 4)R_c]}{R_c^{\alpha + 2}(R_c - R_{sw})^2} , \] (2.66)

\[ B = \frac{\alpha[(\alpha + 1)R_{sw} - (\alpha + 3)R_c]}{R_c^{\alpha + 2}(R_c - R_{sw})^3} , \] (2.67)

\[ C = \frac{1}{R_c^\alpha} - \frac{1}{3}A(R_c - R_{sw})^3 - \frac{1}{4}B(R_c - R_{sw})^4 . \] (2.68)

The expression for the shifted or switched force on particle \( i \) by particle \( j \) for the three non-bonded interaction terms \( V^\alpha_{ij}(r_{ij}) \) is then

\[ F^\alpha_{ij}(r_{ij}) = -\frac{\partial V^\alpha_{ij}(r_{ij})}{\partial r_{ij}} = -c_\alpha \Phi^\alpha_{ij}(r_{ij}) \frac{r_{ij}}{r_{ij}} , \] (2.69)

with

\[ \Phi^\alpha_{ij}(r) = \begin{cases} 
\Phi^\prime_{ij}(r) & r \leq R_{sw} \\
\Phi^\prime_{ij}(r) + S^\prime_{ij}(r) & R_{sw} \leq r \leq R_c \\
0 & r \geq R_c 
\end{cases} , \] (2.70)

and

\[ S^\prime_{ij}(r) = -A(r - R_{sw})^2 - B(r - R_{sw})^3 . \] (2.72)

We note that the switched potential energy determined by Equation (2.62) and forces from Equation (2.69) are only correct for a distance dependence of the potential energy function of the form of Equation (2.60). So, it cannot be used in the presented form if the soft-core interaction or reaction-field forces as defined in GROMOS are to be used.

**Free-energy through one-step perturbation**

The calculation of relative binding free energies of many ligands to a common receptor is of relevance for drug design and screening purposes, and for obtaining a better understanding of interactions governing molecular complexation in general. The one-step perturbation technique allows for the calculation of a great many relative free energies from a single simulation of a (not necessarily physically meaningful) reference state. The idea behind the method is to simulate a judiciously chosen reference compound \( R \) generating an ensemble of structures that contains conformations representative for many physically relevant compounds. The free energy difference between any real ligand \( A \) and the reference compound \( R \) can be obtained from the perturbation formula

\[ \Delta G_{AR} = \Delta G_A - \Delta G_R = -k_BT \ln \left( \frac{e^{-\beta(H_A - H_R)/k_BT}}{R} \right) , \] (2.73)
where the angular brackets indicate the ensemble average of the configurations generated in a simulation of \( R \). \( H_A \) and \( H_R \) are the Hamiltonians for the real compound (\( A \)) and the reference compound (\( R \)), respectively. Because this expression involves the difference between two Hamiltonians, only interactions that differ between compounds \( A \) and \( R \) need to be reevaluated over the ensemble. This allows for the calculation of thousands\(^{94} \) to millions\(^{92} \) of relative free energies from a handful of simulations of reference states \( R \).

The success of the method critically depends on the choice of the reference state \( R \), it should allow wide sampling, but not so wide that insufficient statistics is obtained. One of the key elements that allow wide sampling is the use of soft-core non-bonded interactions, which allows for a spatial overlap between these atoms. In GROMOS96, the soft-core non-bonded interaction was chosen to be of the form\(^{3,4,86} \)

\[
V^{\text{sc}}(r_{ij}) = \frac{4\varepsilon_{ij}\sigma_{ij}^6}{s_{\lambda}(i,j)\lambda^2\sigma_{ij}^6 + r_{ij}^6} \left[ \frac{\sigma_{ij}^6}{s_{\lambda}(i,j)\lambda^2\sigma_{ij}^6 + r_{ij}^6} - 1 \right] \\
+ \frac{q_i q_j}{4\pi\varepsilon_0\varepsilon_1} \left[ \frac{1}{(s_C(i,j)\lambda^2 + r_{ij}^2)^{\frac{1}{2}}} - \frac{1}{s_C(i,j)\lambda^2 + R_{rf}^2} \right] - \frac{1 - \frac{1}{2}C_{rf}}{R_{rf}}. \tag{2.74}
\]

In GROMOS96 the soft-core parameters \( s_{\lambda} \) and \( s_C \) were taken equal for all soft-core atom pairs. In GROMOS05 (MD++ only) \( s_{\lambda}(i,j) \) and \( s_C(i,j) \) are calculated by combining the distinct softness parameter specified per atom, allowing fine-tuning of the reference state, i.e.

\[
s(i,j) = \begin{cases} \\
    \frac{1}{2}(s(i) + s(j)) & s(i) \neq 0, s(j) \neq 0 \\
    s(i) & s(i) \neq 0, s(j) = 0 \\
    s(j) & s(i) = 0, s(j) \neq 0 \\
    0 & s(i) = 0, s(j) = 0.
\end{cases} \tag{2.75}
\]

The GROMOS++ post-processing programs \texttt{pt.top} (to generate a real topology from a topology and a perturbation topology) and \texttt{ener} (to recalculate the interaction energy of specified atoms) or \texttt{m_pt.top} and \texttt{m_ener} (to do the same for multiple physical compounds at the same time), and \texttt{dg.ener} (to calculate the relative free energies) may be used to analyse the reference state ensemble.

### 2.5 Code organisation, implementation

#### 2.5.1 MD engine in FORTRAN: PROMD

The FORTRAN MD engine (PROMD) is an enhancement of the GROMOS96 MD engine. It is written in FORTRAN77 except for the use of include files and macro preprocessing. Macros are in
particular used to get rid of unnecessary features (such as four-dimensional simulation, unused periodicity code or unused perturbation code) so as to improve the performance for specific applications through the use of a specialised code. To facilitate performance tuning, timing routines have been included, and the time spent within various components of the program is reported at the end of each simulation. Major additional algorithmic features (with respect to GROMOS96) have been described in Section 2.3.

2.5.2 MD engine in C++: MD++

The C++ MD engine (MD++) has been written from scratch. The major motivation was to further increase the modularity and therefore the extendability of the MD program. The code is split into two parts, the first one being an MD library containing basic functions necessary to run an MD simulation, the second one being the actual MD program. This second part is very small. It is therefore easy to write other specialised MD programs that make use of a subset of the functions provided in the library or apply them in a different order. The source code of the library is in turn split up into nine different parts: *math, simulation, topology, configuration, algorithm, interaction, io, util* and *check* (represented as C++ namespaces).

- *math* contains classes for vectors, matrices and vector arrays, mathematical operations, physical constants and periodic boundary treatment.
- *simulation* contains the simulation parameters supplied to run an MD or SD simulation or an EM.
- *topology* contains the topology of the simulated system, possibly also including a perturbation topology.
- *configuration* contains the state of a system: its coordinates, velocities, forces, restraints data and so on.
- *algorithm* contains classes that use information from *simulation* and *topology* to act upon a *configuration*. All steps during an MD or SD simulation or EM can be carried out using an *algorithm*.
- *interaction* contains the largest algorithm: the energy, forces and virial evaluation. Here, all interaction terms and their parameters are defined. Because of its size, *interaction* is a separate part, though it formally belongs to *algorithm*. The *interaction* part is further split into *bonded, nonbonded* and *special* interactions.
- *io* contains classes to read in or write out information. All file access is block oriented and human readable.
2.5. Code organisation, implementation

- **util** contains a few extra classes that are necessary to set up a simulation but which do not exactly belong to it. Parsing of command line arguments, generation of initial velocities or setting of debug levels are examples of classes found herein.

- **check** contains test routines. Testing includes the automatic calculation of energies under different conditions as well as the calculation of forces, virial tensor and energy λ-derivatives and their comparison to values obtained by finite difference calculations.

One step of an MD or SD simulation or EM consists of several Algorithms (Figure (2.5.2)) applied to the Configuration in the right order.

```cpp
class Algorithm{
    public:
    Algorithm(string name) : name(name) {}
    ~Algorithm() {}
    virtual int init(Topology & topo,
                     Configuration & conf,
                     Simulation & sim) = 0;

    virtual int apply(Topology & topo,
                      Configuration & conf,
                      Simulation & sim) = 0;

    string name;
};
```

*Figure 2.2: Interface of the Algorithm class.*

The AlgorithmSequence class (Figure 2.5.2) is a container for all these algorithms. When a simulation is set up, they are inserted in the correct order into the AlgorithmSequence. Different groups of algorithms (like temperature coupling algorithms) correspond to the STRATEGY pattern of software engineering. Before the start of a simulation, all algorithms will be initialised (by calling the init() function). During an MD step (AlgorithmSequence::run()), the algorithms are applied (by calling Algorithm::apply()). The forcefield itself is also an algorithm, which, when applied, calculates the energies, forces and virial contribution of all force-field terms for the complete system. The force-field terms themselves are Interaction classes. The Forcefield is therefore a container to store the different Interaction objects (in analogy to the AlgorithmSequence and Algorithm classes). When the force field is applied, it calls calculate_interactions() on all interaction objects. There are distinct interaction
class Algorithm_Sequence : public vector<Algorithm *> {

public:
    Algorithm_Sequence();
    ~Algorithm();

    int init(Topology & topo,
              Configuration & conf,
              Simulation & sim);

    int run(Topology & topo,
             Configuration & conf,
             Simulation & sim);

    Algorithm * algorithm(string name);
};

Figure 2.3: Interface of the Algorithm_Sequence class.

objects for the covalent interactions (bond-length, bond-angle, improper-dihedral and torsional-dihedral interactions), the non-bonded interactions (pairlist construction, long-range interactions and short-range interactions) and the non-physical interactions (atom-position, atom-distance, dihedral-angle, NOE or J-value restraints). It is very easy to add a custom Interaction class to calculate a non-standard interaction.

The classes corresponding to the steps in the MD, SD or EM algorithm are shown in Table 2.5.2 and an overview of the (non-bonded) interaction classes is given in Figure 2.4.

The Nonbonded_Sets contain independent subsets of the non-bonded interactions. Their calculate_interactions() method may be called in parallel (using either shared or distributed memory parallelisation). The Nonbonded_Sets share (through the Nonbonded_Interaction) a pairlist construction algorithm, which they call to create the part of the complete pairlist relevant to them. These different parts of the pairlist stay together with the Nonbonded_Set and need never be assembled into the complete pairlist. To gain flexibility, the calculation of the individual atom - atom pair interaction is further split up into a Nonbonded_Outerloop (loops over the atom - atom pairs), a Nonbonded_Innerloop (prepares the parameters necessary to calculate the interaction) and a Nonbonded_Term (calculates the atom - atom pair interaction energy, force and virial contribution). The Storage class provides directly accessible (local) memory for each Nonbonded_Set.
2.5. Code organisation, implementation

1. Write position and velocity components.
2. Remove centre of mass motion.
3. Calculate (unconstrained) forces and energies from the potential energy function (using nearest image convention in case of periodic boundary conditions). Save these.
4. Satisfy position constraints.
5. Update the velocities using the leap-frog scheme.
6. Apply temperature coupling (weak coupling or Nose-Hoover(-chains)).
7. Update the positions using the leap-frog scheme.
8. Satisfy distance constraints (using SHAKE, M-SHAKE or LINCS).
9. Calculate temperature(s).
10. Calculate pressure.
11. Apply pressure scaling (weak coupling).
12. Update lambda and topology for slow-growth simulations.
13. Calculate total energies, averages and fluctuations. Save these.

Table 2.3: Classes corresponding to MD algorithm steps.

Efficiency

The main goal for writing a new C++ MD engine was to further improve on modularity (using some object-oriented features) and extendability (using clear and common interfaces between the modules). Nevertheless, a simulation code has to be reasonably efficient to be of practical use. The complete code is written in standard C++98, no language extensions or machine-specific parts are used anywhere, resulting in a highly portable program. This means that the compiler has to do all machine specific optimisations. We believe that the absence of any machine specific parts of code, which require duplication to be able to run on different machines, facilitates future
Figure 2.4: Illustration of the Interaction classes in MD++. The red arrows denote a is-a relationship, the black arrows has-a. All Interaction classes inherit from Interaction and, therefore, can be stored in the Forcefield, which is a vector of Interaction classes. The NonbondedInteraction consists of a PairlistAlgorithm (either a StandardPairlistAlgorithm or a Grid-BasedPairlistAlgorithm) and (depending on parallelisation) one or more NonbondedSets. Those, in turn, consist of Storage (to locally store forces, energies, virial tensor and pair lists) and an Outerloop (to calculate the interactions). The Outerloop relies on the Innerloop and on Term to calculate the interactions.

modification. Furthermore, current compilers are getting ever better at producing fast programs, making use of the specific features available on the machine.

In the inner loops of the interaction calculation, templates are used to generate specialised code. There are, for instance, specialised periodicity classes for the different implemented types of periodic boundary conditions (vacuum, rectangular, truncated octahedral and triclinic). The Innerloop methods are called with the boundary type as a template argument. Thus the compiler will generate a different specialised version of the inner loops for different boundary conditions.
2.5. Code organisation, implementation

automatically. In the same manner, the interaction function term of the non-bonded interaction can also be chosen (e.g. with or without switching function for non-bonded interactions) without any if statement required in the compiled inner loop. An example code fragment is shown in Figures 2.5 and 2.6. The same technique is used to implement perturbation simulations and different definitions of the virial tensor.

Some algorithms do rely on information from the previous integration step. To help implementing those kind of algorithms, the complete current and old state (positions, velocities, forces, energies, restraint and constraint data, averages, and so on) of the simulation are stored. During the leap-frog algorithm, the current state becomes the old state and the updated information is stored in the new current state. This transfer is done by a simple (and fast) pointer exchange. This slightly increases memory usage (but the required space is still small compared to that used to store the pairlists).

A comparison of the efficiency of the C++ code with respect to the GROMOS96 MD engine (in FORTRAN) is given in Table (2.2). This comparison shows that MD++, using the standard pairlist algorithm (std) is approximately a factor two slower than GROMOS96 (standard pairlist algorithm), improved algorithms (like a grid-based pairlist construction) may have a large impact on the performance reducing the time spent to two third for the membrane and even by a factor of three for the protein system. Note that the optimised pairlist construction algorithm implemented in the FORTRAN MD engine (PROMD) only benefits from more efficient processor cache usage but still scales as $O(N^2)$ with system size. Still, for the systems tested here, it achieves equal overall efficiency as the $O(N)$ scaling grid-based pairlist algorithm in MD++. The future will show whether the improved extendability of MD++ will, through improved algorithms, lead to a faster C++ code than the FORTRAN (PROMD) code or whether slightly inferior performance is the price to pay for a more structured code layout.

Debugging information

It is often difficult to figure out what is going on during an MD or SD simulation or an EM and many users tend to use the program as a black box. MD++ tries to improve this situation by enabling the user to select a tuneable amount of information to be printed out during the simulation. Every (output or debugging) message is associated with a debugging level, and the message is printed only if the requested debugging level is high enough. Additionally, every code section belongs to a module and a submodule. Different debug levels can be specified for all combinations of modules and submodules. In that way, fine grained control is achieved on how much information from which part of the MD++ code should be printed.

Parallelisation

Computationally, the interaction calculation is by far the most expensive part of an MD or SD simulation or an EM, while the non-bonded interactions constitute the bulk of the effort. Again,
enum boundary_type {vacuum, rectangular, triclinic};
template<boundary_type boundary>
class Periodicity;

template<>
class Periodicity<vacuum>{
    public:
    void nearest_image(Vec const & v1, Vec const & v2, 
                       Vec & v3);
};
template<>
class Periodicity<rectangular>{
    public:
    void nearest_image(Vec const & v1, Vec const & v2, 
                       Vec & v3);
};
// and similar ones for triclinic or 
// truncated octahedral boundary conditions

template<boundary_type boundary>
class Interaction{
    public:
    virtual int calculate_interactions(
        Topology const & topology, 
        Configuration & configuration, 
        Simulation const & simulation){
        Vec v;
        Periodicity<boundary> 
            periodicity(configuration.current().box);
        periodicity.nearest_image(
            configuration.current().pos(0), 
            configuration.current().pos(1), 
            v);
        // and so on
    }
};

Figure 2.5: Specialised code generation using templates.
2.5. Code organisation, implementation

```c
int main(int argc, char **argv)
{
    Interaction<rectangular> interaction;
    interaction.calculate_interactions(
        topology, configuration, simulation);

    return 0;
}
```

Figure 2.6: Using the periodicity class.

MD++ is focused on achieving parallelisation without complicating the code. The non-bonded interaction is split up into NonbondedSets, each containing its own storage space for a pairlist, energies, forces, and virials. In this way, the standard code is ready for shared and distributed memory parallelisation without any need for code duplication. If the system is using distributed memory, the (updated) positions have to be copied from the master to all other processes before the next interaction calculation. While composing the pairlist in parallel, only a subset of atoms is considered per process, so that each processor creates its own partial and local pairlist. The interactions are calculated from this partial pairlist and stored in local arrays. This ensures synchronisation for shared memory machines and replicated data parallelisation for distributed memory systems. After the partial interaction calculations have finished, the energies, forces, and virials of all non-bonded sets are summed up and stored in the Configuration of the master process.

MD++ can use OpenMP\(^\text{\textcopyright 1997}\) for shared memory and MPI\(^\text{\textcopyright 1998}\) for distributed memory parallelisation. Reasonable parallelisation (using a small number of parallel processes) can be achieved with only a few lines of code (almost) completely separate from the non-bonded routines (see Table (2.2)). In the pairlist generation, each process only creates a partial pairlist for specified (central) atoms (or grid-cells if a grid-based pairlist construction is used). These partial pairlists are then used in the force calculation within the same process. This way, only the final, total forces need to be summed up over all the processes. Using these forces, the master process performs the integration step and then the positions of the atoms are updated in all processes and the next force calculation (using the previously generated pairlists or doing a complete update) can start.
2.5.3 Analysis modules: GROMOS++

All the FORTRAN analysis programs of GROMOS96 have been rewritten in C++. They accept a standard set of command line arguments to specify input. It is easy to add new analysis programs using the functionality provided within the GROMOS++ library. Following is a short description of the existing programs.

Setup of simulations (pre-processing)

- `make.top` builds a topology from a building block sequence.
- `com.top` combines two topologies.
- `con.top` converts topologies to a different force-field parameter set.
- `red.top` reduces topologies by specified parts.
- `pt.top` combines topologies with perturbation topologies to produce new topologies or perturbation topologies.
- `pert.top` creates a perturbation topology to perturb specified atoms to dummies.
- `check.top` checks topologies for common mistakes.
- `pdb2g96` converts a pdb (Protein Data Bank) structure into GROMOS coordinates.
- `build.box` builds a simulation box containing $N$ molecules at a specified density.
- `ran.box` builds a simulation box containing $N$ molecules at a specified density, placing and orienting them randomly.
- `bin.box` builds a simulation box containing a binary mixture at a specified density.
- `sim.box` puts a simulation box around a molecule and fills it with solvent molecules from an equilibrated solvent configuration.
- `ran.solvation` builds a simulation box around a molecule and fills it randomly with solvent molecules.
- `check.box` checks box properties (size).
- `copy.box` multiplies a box in any direction.
- `explode` increases inter-molecule distances to vacuum conditions.
- `cry` applies rotations and translations to a system to create a crystal unit cell.
2.5. Code organisation, implementation

- ion replaces a specified number of solvent molecules by ions.
- gch generates hydrogen atom coordinates for a molecule.
- gca generates atomic Cartesian coordinates from a set of internal coordinates.
- mk_script prepares an MD, SD or EM job script.

Analysis of trajectories (post-processing)

- tstrip removes solvent from a trajectory.
- filter filters out specified atoms from a trajectory.
- cog calculates centre of geometries for specified atoms.
- tser calculates time series of specified properties (distances, angles, torsions, order parameters, etc.).
- tcf calculates time correlation functions of time series.
- dist calculates distributions of specified properties.
- ditrans monitors dihedral-angle transitions.
- propertyrmsd calculates root-mean-square differences for a set of properties.
- dipole calculates dipole moments with respect to the centre of molecules.
- rmsd calculates atom-positional root-mean-square differences between structures.
- rmsf calculates atom-positional root-mean-square fluctuations for specified atoms.
- ene_ana calculates averages, fluctuations and error estimates for energies, pressure and volume.
- epsilon calculates the dielectric permittivity for liquids.
- visco calculates the shear viscosity of liquids.
- rgvr calculates the radius of gyration.
- rdf calculates the radial distribution function for selected atoms.
- mdf gives the time series of the closest particle index to a selected atom.
- m_widom performs widom particle insertion.
• sasa calculates the solvent accessible surface area (SASA) for a specified part of a molecule.
• hbond analyses hydrogen bonding.
• dssp analyses secondary structure elements.
• prep_noe prepares for an NOE calculation.
• noe calculates NOE distances.
• post_noe analyses NOE distances.
• oparam calculates order parameters for lipids in membranes.
• nhoparam calculates N-H order parameters.
• diffus calculates the diffusion coefficient of specified atoms.
• rmsdmat calculates the rmsd between all structure-pairs in a trajectory.
• cluster analyses an rmsd-matrix to separate the structures into clusters.
• postcluster analyses the cluster output for lifetimes, folding pathways and central-member structures.
• iondens calculates ion densities.
• edyn performs an essential dynamics analysis.
• rot_rel calculates the rotational relaxation time for solvent molecules.
• ener calculates any energy for a system.
• espmap calculates the vacuum electrostatic potential on a grid around selected molecules of a given configuration from the partial charges in the topology.

Miscellaneous

• frameout converts trajectories into other formats or extracts snapshots from trajectories.
• inbox puts the solute into the centre of the box.
• atominfo prints (topological) information on specified atoms.
• shake_analysis analyses a specified configuration.
• cmt_list lists atoms within a specified distance from a given atom.
2.6 Examples of application

2.6.1 Local-elevation simulation of glucose

The technique of local-elevation (LE) MD has been developed to enhance the searching of the configurational space of a molecule by progressively elevating the local potential energy of the configurations that are visited during an MD trajectory\textsuperscript{99}. The total potential energy function consists of two terms, the standard physical terms $V_{\text{phys}}(\mathbf{r}(t))$ and the local-elevation term $V_{\text{LE}}(\mathbf{r}(t), t)$ which depends also explicitly on the time $t$. When the molecule is trapped in a low energy basin of the potential energy hypersurface, the LE algorithm gradually elevates the bottom of this basin using additional Gaussian-shaped energy functions, which will eventually force the molecular system out of the basin into a neighbouring basin of the energy hypersurface. In this way, the energy surface is much more efficiently sampled than using standard MD. For low-dimensional systems LE-MD will lead to a flat potential energy as soon as all parts of the LE configuration space have been visited\textsuperscript{99}. If

$$V_{\text{phys}}(\mathbf{r}(t)) + V_{\text{LE}}(\mathbf{r}(t), t)$$

is flat after a (long) time $t_l$, by construction of the local-elevation potential energy term, $V_{\text{LE}}(\mathbf{r}, t_l)$ represents the negative of the free-energy surface or potential of mean force for the LE degrees of freedom of the molecule.

LE-MD was already implemented in GROMOS96\textsuperscript{3,4}. Here, we illustrate its sampling efficiency using as an example the conformational sampling of a glucose molecule (Figure (2.7)) solvated in SPC water\textsuperscript{100}.

![Glucose molecule with atom numbering](image)

**Figure 2.7:** Glucose molecule with atom numbering
The time course of the six exocyclic torsional dihedral angles of the sugar ring are shown for a standard MD and for LE-MD simulation in Figure (2.8).

In the standard MD at 300K and 1 atm, no conformational transitions are observed on a 1 ns timescale, while the LE-MD simulation with an energy weight factor that is raised by $2kJmol^{-1}$ every time a configuration is revisited already leads to a first conformational transition after 200 ps. After 500 ps many transitions are observed, indicating that the potential energy surface Equation (2.76) is becoming flat, the free-energy surface can then be obtained in the form of $-VLE(r, t > 1000ps)$.

### 2.6.2 Replica-exchange simulation of butane

500 butane molecules, all in trans-configuration have been simulated at 273 K. The force constant of the torsional angle has been increased by a factor 3. The time to reach the equilibrium state of *gauche* and *trans* butane can be determined by monitoring the width of the torsional-angle distribution. The potential energy barrier between the *trans* and the *gauche* configuration is too high ($\approx 18kJ/mol$) to overcome at 273 K. REMD is applied to increase the sampling of configurational space at 273 K. To this effect, 11 replicas of the system with changed torsional-angle force constants (scaled by 1.0, 0.9, 0.8, 0.75, 0.7, 0.65, 0.6, 0.55, 0.5, 0.45 and 0.4, respectively) were simulated simultaneously. Every 0.5 ps, an exchange between two neighbouring replicas was attempted. Figure (2.9) shows the path in $\lambda$ - space for all replicas. The overall exchange probability during the simulation was 0.25. The time series of the root-mean-square deviation (rmsd) from the average of the torsional angle of all butane molecules is depicted in Figure (2.10). The larger the rmsd is, the more torsional angle transitions from *trans* to *gauche* have happened. Through replica exchange, also the simulation running at $\lambda = 0$ contains butane molecules in *gauche* conformation, unlike the simulation carried out without replica exchange. Note that the rmsd is dependent on the width of the valleys, so it is dependent on the force constant. This in turn means that the equilibrium value of the rmsd is different for the different replicas.

### 2.6.3 Coarse-grained simulation of alkanes

Coarse-grained (CG) models allow for much more efficient sampling of the molecular configurational space than atomic-level (AL) models (at the expense of loss of atomic detail). Yet a CG model should be able to reproduce the properties of an AL model, assuming that the latter is correct. To illustrate this requirement for CG models, some properties (conformational distributions and configurational entropies) of n-alkanes in the liquid phase have been compared. The main results are summarised here in the context of hexadecane, where the AL model was the standard GROMOS 45A3 force field and the CG model the one discussed before. For the AL model 128 hexadecane molecules were simulated at 323 K and 1 atm in a periodic box over 25 ns. For
2.6. Examples of application

Figure 2.8: Time course of the six torsional dihedral angles of the glucose ring as obtained from standard MD (A) and local-elevation (LE-) MD (B). The LE weight factor was $E_{\phi}^{le} = 2 \text{kJ mol}^{-1}$ and the four torsional angles $C(1)-C(2)-C(3)-C(4)$, $C(2)-C(3)-C(4)-C(5)$, $C(3)-C(4)-C(5)-O(5)$ and $C(5)-O(5)-C(1)-C(2)$ were chosen as LE degrees of freedom.
the CG model 512 molecules were simulated over 1000 ns under the same conditions. In the AL model, a hexadecane molecule consists of a linear chain of 16 united atoms. In the CG model, four united atoms are represented by one bead, so that hexadecane consists of 4 beads. In order to compare AL configurations of united atoms with CG configurations of beads, the AL trajectories were mapped to the CG level by considering only the centres of mass of the four united atoms that represent one bead at the CG level. This mapping of the atomic level onto the coarse-grained level is indicated by the symbol MAP.

Figure (2.11) shows the distribution of the values of the two pseudo bond angles and the one pseudo torsional angle of the hexadecane molecules at the CG level for the MAP (grey) and CG (black) trajectories.

The difference in torsional-angle distribution can be explained from the absence of torsional potential energy terms in the CG model\(^3\). Table (2.4) shows the configurational entropies of the four united atom fragments of the hexadecane molecules at the atomic level (AL) and at the CG
Figure 2.10: *Simulation of liquid butane, starting from an all trans configuration of the torsional angle.* The time series of the root-mean-square deviation (rmsd) from the average torsional angle is shown. The bold black line depicts the rmsd in the standard MD simulation. No broadening of the distribution is visible. The bold red line denotes the rmsd of the replica at $\lambda_i = 0.0$ (corresponding to the standard MD simulation). Clearly, the relaxation towards the equilibrium state is much faster using the replica-exchange method than in the standard MD simulation. The other lines denote the other replicas ($0.0 < \lambda_i \leq 1.0$), many of them reaching their equilibrium state already after about 50 ps.

At the CG level (MAP), together with those obtained from the CG simulations (CG). At the CG level the configurational entropies of MAP and CG models agree very well, to within 2 Jmol$^{-1}$K$^{-1}$.

These data illustrate that the CG model$^{83}$ is able to reproduce the properties of the GROMOS AL model rather well.
Figure 2.11: Bond-angle (A-B-C, B-C-D) and torsional dihedral angle (A-B-C-D) distributions at the coarse-grained level. Grey: A-D are centres of mass of fragments consisting of four united atoms as obtained from AL trajectories. Black: A-D are beads of the CG model obtained from CG simulations.
2.6. Examples of application

<table>
<thead>
<tr>
<th>fragment</th>
<th>AL</th>
<th>MAP</th>
<th>CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>211</td>
<td>133</td>
<td>131</td>
</tr>
<tr>
<td>B</td>
<td>209</td>
<td>111</td>
<td>110</td>
</tr>
<tr>
<td>C</td>
<td>209</td>
<td>111</td>
<td>110</td>
</tr>
<tr>
<td>D</td>
<td>211</td>
<td>133</td>
<td>131</td>
</tr>
</tbody>
</table>

Table 2.4: Configurational entropy (in J K\(^{-1}\) mol\(^{-1}\)) of the four (A-D) hexadecane fragments that correspond to the four beads of the coarse-grained (CG) model for hexadecane in the liquid phase. AL: atomic-level entropies; MAP: fragment entropies from the AL trajectories; CG: bead entropies from the CG trajectories.

2.6.4 One-step perturbation calculations on the free energy of ligand binding to the estrogen receptor

The one-step approach to calculate relative free energies of complexation or ligand binding is particularly efficient when many structurally not too different ligands are to be considered. Previously\(^9^1\), the Gibbs free energy of binding for 17 polychlorinated biphenyls to the estrogen receptor were calculated from two MD simulations of an unphysical reference compound, one when bound to the protein and one free in solution. Here, the efficiency of the one-step technique is illustrated by calculating more than 2000 binding free energies from the two simulations.

Figure (2.12) shows the biphenyl ligand with the 9 atoms that are made soft atoms in the unphysical reference state. At these nine soft sites, five different real substituents (H, F, Cl, Br and CH\(_3\)) can be placed, yielding $5^9 - 1 = 1953124$ relative binding energies for the ligands. Here, we calculated free energies for all possible polyfluorinated, polychlorinated, polybrominated, and polymethylated ligands (in total $4 \cdot 2^9 - 1 = 2047$ relative free energies) from one simulation of the free ligand in water and one bound to the estrogen receptor. For every class of substituted biphenyl ligands, the three best binding structures are depicted in Figure (2.12). (We note that the binding affinity of the polybrominated biphenyls might be underestimated by the choice of the reference state: the soft-core atoms chosen have smaller van der Waals radii than the Bromine atoms.)

This application illustrates the efficiency of the one-step perturbation technique for screening purposes in drug design.

2.6.5 Other applications

GROMOS can be used for molecular modelling of any type of molecular system. Below, a number of applications are mentioned, which, for convenience have mainly been taken from our own more recent work.
The structural stability of proteins, peptides, sugars, and DNA as function of their composition, chain lengths and solvent environment or temperature and pressure can be studied. Solvation, both in pure solvents and in mixtures can be investigated in atomic detail. Motional properties, NMR coupling constants and dielectric relaxation times can be analysed. J-coupling constants, NOE's and NMR order parameters and CD spectra can be compared to experimental values. GROMOS can also be used for structure refinement of biomolecules based on NMR data. Molecular host-guest complexes can be studied in terms of structural properties and free energy and entropy of binding. Polypeptide (un)folding equilibria can be simulated in atomic detail. Biochemical reactions can be mimicked in QM/MM simulations, in which interfaces to quantum chemistry software have to be used.

A variety of types of molecules have been simulated: proteins, DNA, RNA, saccharides,
lipids and a range of solvents: water, DMSO, methanol, chloroform, carbon tetrachloride, acetone-nitrile and mixtures of these and other cosolvents such as urea. Also membranes and micelles have been simulated using GROMOS.

2.7 Conclusions

The GROMOS software for biomolecular simulation has been extended with new functionality and extended analysis possibilities and put partially into C++, which makes extension of functionalities easier. GROMOS05 comes with the latest thermodynamically calibrated GROMOS force-field parameter sets 45A3/4 and 53A5/6, which are suitable for a broad range of molecular systems. The source code of GROMOS is obtainable for a nominal fee and should allow both methodological investigations and structural, dynamical and energetic explorations of biomolecular systems which may lead to an enhanced understanding of the properties of such systems.

2.8 Acknowledgements

Financial support by the National Center of Competence in Research (NCCR) Structural Biology of the Swiss National Science Foundation (SNSF) is gratefully acknowledged.
2.9 Bibliography


2.9. Bibliography


2.9. Bibliography


2.9. Bibliography


[95] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns* (Addison-Wesley, 1995).


Chapter 2. The GROMOS software for biomolecular simulation: GROMOS05


2.9. Bibliography


2.9. Bibliography


Chapter 3

On searching in, sampling of, and dynamically moving through conformational space of biomolecular systems: a review

3.1 Summary

Methods to search for low-energy conformations, to generate a Boltzmann-weighted ensemble of configurations, or to generate classical-dynamical trajectories for molecular systems in the condensed liquid phase are briefly reviewed with an eye to application to biomolecular systems. After having chosen the degrees of freedom and method to generate molecular configurations, the efficiency of the search or sampling can be enhanced in various ways: (i) efficient calculation of the energy function and forces, (ii) application of a plethora of search enhancement techniques, (iii) use of a biasing potential energy term, and (iv) guiding the sampling using a reaction or transition pathway. The overview of the available methods should help the reader to choose the combination that is most suitable for the biomolecular system, degrees of freedom, interaction function, and molecular or thermodynamic properties of interest.
3.2 Introduction

Computer modeling of (bio)molecular systems has become a standard technique to study and describe the properties and behaviour of such systems in terms of interactions between atoms or electrons of atoms. Although quantum mechanics governs these interactions, it can only be used to model a very limited number of degrees of freedom of a biomolecular system due to the complexity of the algorithms to solve the (time-dependent) Schrödinger equation. Leaving processes such as electron or proton transfer or processes that involve making or breaking of covalent bonds aside, classical mechanics offers a good approximation of quantum mechanics, e.g. for processes such as polypeptide folding, molecular complexation, partitioning of molecules between different environments and the formation of molecular aggregates (e.g. membranes) out of mixtures. Here we only consider molecular systems that are described in terms of a classical Hamiltonian

$$\mathcal{H}(\mathbf{p}, \mathbf{q}) = \mathcal{K}(\mathbf{p}, \mathbf{q}) + \mathcal{V}(\mathbf{p}, \mathbf{q}),$$

(3.1)

which depends on the $\mathbf{q} \equiv (q_1, q_2, \ldots, q_{N_{df}})$ generalised coordinates and $\mathbf{p} \equiv (p_1, p_2, \ldots, p_{N_{df}})$ conjugate momenta of the chosen $N_{df}$ degrees of freedom. The kinetic energy term is denoted by $\mathcal{K}(\mathbf{p}, \mathbf{q})$ and the potential energy one by $\mathcal{V}(\mathbf{p}, \mathbf{q})$. The classical-mechanical equations of motion are then

$$\frac{d}{dt} q_i = \frac{\partial \mathcal{H}(\mathbf{p}, \mathbf{q})}{\partial p_i}, \quad i = 1, 2, \ldots, N_{df},$$

$$\frac{d}{dt} p_i = -\frac{\partial \mathcal{H}(\mathbf{p}, \mathbf{q})}{\partial q_i}, \quad i = 1, 2, \ldots, N_{df}. \tag{3.2}$$

When using Cartesian coordinates, $\mathbf{q} \equiv \mathbf{x}$, and assuming that the potential energy is independent of the momenta, one has

$$\mathcal{K}(\mathbf{p}, \mathbf{x}) = \sum_{i=1}^{N_{df}} \frac{p_i^2}{2m_i}, \tag{3.3}$$

and Equations 3.2 reduce to Newton’s equations of motion

$$m_i \frac{d^2 x_i}{dt^2} = -\frac{\partial}{\partial x_i} \mathcal{V}(x_1, x_2, \ldots, x_{N_{df}}), \tag{3.4}$$

where we have indicated the mass governing the motion of the $i$-th degree of freedom by $m_i$. The interaction function

$$\mathcal{V}(\mathbf{x}) \equiv \mathcal{V}(x_1, x_2, \ldots, x_{N_{df}}) \tag{3.5}$$

is an effective interaction: it describes the interaction between explicitly treated degrees of freedom averaged over the omitted atomic or electronic degrees of freedom.

Because biomolecular modeling involves microscopic systems at non-zero temperatures $T$, the basic theory to describe such a system is quantum or classical statistical mechanics. Consequently, the state of a biomolecular system is characterised by a statistical-mechanical ensemble
of configurations. At fixed particle number, volume and temperature this is a canonical ensemble, in which the weight of a molecular or system configuration is given by the Boltzmann factor

\[ e^{-\mathcal{V}(\mathbf{x})/k_B T}, \]  

where \( k_B \) denotes Boltzmann's constant. This implies that the equilibrium properties of the system are determined by those parts of configuration space, for which \( \mathcal{V}(\mathbf{x}) \) is minimal. Therefore, one of the basic challenges to biomolecular modeling is to develop methodology to efficiently search the biomolecular energy surface \( \mathcal{V}(\mathbf{x}) \) for regions of low energy. The statistical-mechanical nature of this search problem implies that it cannot be reduced to the problem of finding the global (energy) minimum of the multi-dimensional function \( \mathcal{V}(\mathbf{x}) \). Statistically-mechanically the free energy

\[ F = U - TS, \]

composed of an energetic contribution \( U \) and an entropic contribution \(-TS\), is minimal, not the energy \( U \). The entropy is a measure of the extent of configurational space \( (\mathbf{x}) \) accessible to the molecular system at a given temperature \( T \). Figure (3.1) illustrates that lowest energy does not necessarily mean lowest free energy. Two parts of configurational space \( \mathbf{x}_1 \) and \( \mathbf{x}_2 \) may have \( U(\mathbf{x}_1) << U(\mathbf{x}_2) \), whereas \( F(\mathbf{x}_1) > F(\mathbf{x}_2) \) due to \( S(\mathbf{x}_1) << S(\mathbf{x}_2) \) at the given temperature \( T \). This means that searching for and finding the global energy minimum for a biomolecular system is meaningless when its entropy accounts for a sizable fraction of its free energy.

When considering methods to generate molecular configurations we distinguish three types based on the characteristics of the set of generated configurations:

1. Methods that generate a series of non-related low-energy configurations.
2. Methods that generate a properly (Boltzmann) weighted set of configurations.
3. Methods that generate a (classical) dynamical trajectory of configurations, which are moreover properly (Boltzmann) weighted.

Methods of type 1 should only be used for zero entropy systems, whereas methods of types 2 and 3 yield proper ensembles, so can be used to compute thermodynamic and other equilibrium properties. Only methods of type 3 yield information on dynamical properties of the system.

Generally, modeling of a molecular system involves four choices.

1. Which degrees of freedom are explicitly modeled, i.e. treated in expressions (3.5) and (3.6).
2. Which interaction function or force field \( \mathcal{V}(\mathbf{x}) \) is used to calculate the potential energy of the system and the forces along the explicitly treated degrees of freedom.
3. Which algorithm is used to search for those parts of configurational space for which $V(x)$ is minimal (type 1), or to sample (type 2) or simulate (type 3) the motion along the degrees of freedom.

4. How are the spatial boundaries of the system modeled and which thermodynamic boundary conditions are used.

Biomolecular systems have a density comparable to solids or liquids, but lack the symmetry ordering of the former. They constitute a many-particle system for which no simple reduction to a few degrees of freedom is possible: one is faced with an essential many-particle problem, the solution of which can only be adequately described by numerical simulation. The four choices mentioned all have an impact on the accuracy and efficiency of the modeling. It is the purpose of this article to consider the choices to be made from the viewpoint of accurately and efficiently generating low-energy configurations or ensembles or trajectories for biomolecular systems. Because of the great variety in methods and applications in the literature, we only classify and mention the available methods, with references, and do not review their applications. The classification given may help the reader to find his or her way in the jungle of methods and to choose a combination of methods and techniques that suits his or her purpose best.
3.3 Choice of degrees of freedom

Generally, biomolecular systems are composed of more atoms than can be reasonably modeled on a computer. Depending on the property of interest, the essential degrees of freedom are to be identified and explicitly treated. The remaining ones are then omitted and their effect upon the interaction between or along the explicitly treated ones is included in an averaged manner in the function $V(x)$. For example, in structure refinement of proteins, solvent degrees of freedom are generally omitted\(^1\), although it is experimentally known that protein structure is sensitive to solvent composition. Polypeptide folding or protein-ligand complexation is sometimes modeled without explicitly treating solvent degrees of freedom\(^3\)\(^4\). This enables fast simulation, because the number of solvent degrees of freedom is generally much larger than that of the solutes, but limits the accuracy\(^5\) and applicability of such implicit solvation models. For example, complex enthalpy-entropy compensation effects cannot be captured\(^6\).

On the other hand, implicit treatment of aliphatic hydrogen atoms by using united \(-CH_n-\) (\(n = 1, 2\)) and \(-CH_3\) atoms in simulation of systems that contain many of such moieties, like lipid mixtures and membranes, saves easily a factor of four to nine in computing effort, which is dominated by the computation of non-bonded forces. Yet, no loss in accuracy is observed when comparing properties calculated using all-atom versus united-atom models\(^7\)\(^8\). The use of united atoms is an example of the technique of coarse-graining: groups of atoms, molecules or fragments of molecules are treated as single particles or beads, whose motion is simulated using a single coarse-grained (CG) force field describing inter-bead interactions. When the energy function $V(x)$ of such a coarse-grained model is chosen to be smooth and short-ranged, the efficiency of coarse-grained simulations can be orders of magnitude higher than the corresponding fine-grained (FG) simulations, be it at the expense of the loss of atomic detail and some accuracy\(^9\)\(^-\)\(^13\). Recently, it has been proposed to combine fine-grained and coarse-grained models in one simulation, while the contribution of the two grain levels to the interaction between the atoms or beads is governed by a grain level parameter $\lambda$. This allows for a continuous switching between grain levels, which can in turn be exploited in the replica-exchange technique to enhance the sampling at the various $\lambda$-values\(^14\)\(^15\).

The performance of a CG model in practical applications depends on the chosen coarse-graining procedure: (i) the model resolution (how many FG particles are mapped onto one CG bead), (ii) the mapping procedure (how the CG bead positions are defined in terms of the FG atom positions), (iii) the form of the energy function $V(x)$ of the CG Hamiltonian, and (iv) the experimental and / or FG simulation properties against which the CG model parameters were optimised.

The number of degrees of freedom to be simulated can also be reduced by constraining those which are characterised by high-frequency motions that are not influencing the properties of interest. For molecular systems one may think of bond-length and bond-angle degrees of freedom\(^16\). Holonomic (time-independent) constraints can be implemented in two ways: (i) by for-
mulating Lagrange equations of motion in generalised (e.g. torsional angle coordinates\textsuperscript{17,18}, or (ii) by formulating these equations in Cartesian coordinates (e.g. using Newton's equations (3.4)) and then using Lagrange multipliers to satisfy the constraints for each configuration generated\textsuperscript{19,20}. When considering branched polymers, the choice of internal coordinates (bond lengths, bond angles, and torsional angles) to serve as generalised coordinates seems to be natural, because they allow for constraining bond lengths and angles by simply omitting them from the equations of motion. However, the equations of classical dynamics (3.2) expressed in internal, generalised coordinates $q = \Theta$

$$\sum_{j=1}^{N_{df}} a_{ij} \frac{d^2 \theta_j}{dt^2} = -\frac{\partial}{\partial \theta_i} \mathcal{V}(\theta_1, \theta_2, \ldots, \theta_{N_{df}}) - \sum_{j=1}^{N_{df}} b_{ij} \left( \frac{d}{dt} \theta_j \right)^2$$

$$- \sum_{j=1}^{N_{df}} \sum_{k=1}^{N_{df}} c_{ijk} \left( \frac{d}{dt} \theta_j \right) \left( \frac{d}{dt} \theta_k \right), \quad i = 1, 2, \ldots, N_{df} \quad (3.8)$$

are considerably more complex than when expressed in Cartesian coordinates, Equation (3.4). They contain two additional summations over the number of degrees of freedom and two additional quadratic (i.e. non-linear) terms in the generalised velocities. Equation (3.8) has been presented in different forms\textsuperscript{1,2,17,18,21-25}, and the coefficients $a_{ij}, b_{ij},$ and $c_{ijk}$ depend on the atomic masses and the molecular topology of the polymer considered. Simulation of a protein through Equation (3.8) requires a much larger computational effort than simulation through Equation (3.4), since at each time step a set of $N_{df}$ non-linear equations is to be solved. One iteration to this end may take as much computational effort as the calculation of all forces and energies, thereby doubling the overall computational expense\textsuperscript{2}. Therefore, the use of Cartesian coordinates, i.e. Newton's equations of motion in combination with Lagrange multipliers to impose constraints is recommended\textsuperscript{26}.

An extension of the concept of a (hard) constraint is a flexible, or soft or adiabatic constraint, in which the length of a constrained distance is not a constant through the simulation, but varies per time step without involving kinetic energy\textsuperscript{27-29}. This eliminates the high-frequency motions in the system, while keeping the constrained degrees of freedom flexible.

The number of degrees of freedom can also be kept low by choosing appropriate periodic boundary conditions. When simulating a spherical solute, use of a more spherically shaped configurational periodic box instead of the standardly used cubic or rectangular periodic box may considerably reduce the number of solvent molecules needed to fill the space left after insertion of the solute in the box. For a spherical solute the number of atomic degrees of freedom can be reduced by at least one quarter in this way\textsuperscript{30}.
3.4 Types of methods to search, sample or dynamically move through configuration space

A variety of search, sampling or simulation methods is available, each with its particular strengths and weaknesses, depending on (i) the form of the function $V(x)$, and (ii) the number and types of degrees of freedom of the system. These methods are based on the use of molecular coordinates $q$ or $x$ as variables. For methods that use as variables other quantities than molecular coordinates we refer to Section 3.9. Two basic types of methods can be distinguished, systematic search and heuristic search.

Systematic or exhaustive search methods scan the complete or a significant fraction of the configuration space of the molecular system. Particular subspaces can be excluded from the search without loss in the quality of the solution found, thanks to rigorous arguments that these subspaces can not contain the desired solution. Such arguments are based on a priori knowledge, often of physical or chemical nature, about the structure of the space or energy function or hypersurface to be searched. Systematic search techniques can only be applied to small molecules involving only a few degrees of freedom, because of the exponential growth of the required computing effort as function of the number of degrees of freedom included in the search.

Heuristic search methods, although visiting a tiny fraction of the configuration space, aim at generating a possibly representative (in the Boltzmann weighted sense) set of system configurations. These methods may generally be divided into two or three types.

1. Non-step methods, in which a series of system configurations is generated, which are independent of each other. One example is the so-called distance geometry metric matrix method, which, for a search problem that can be cast into a distance based form, generates, at least in principle, an uncorrelated series of random configurations. Another example is based on the technique of threading, in which linear combinations of parts of protein structures as obtained from a protein structure data bank are used to generate novel possible protein structures.

2. Step methods that build a complete molecular or system configuration from configurations of fragments of the molecule or system in a step-wise manner. Examples are the build-up procedure of Scheraga, combinatorial build-up methods that make use of dynamic programming techniques and Monte Carlo (MC) chain growing methods, such as the so-called configurational bias Monte Carlo (CBMC) technique.

3. Step methods, such as energy minimisation (EM), Metropolis Monte Carlo (MC), molecular dynamics (MD) and stochastic dynamics (SD), that generate a new configuration of the complete system from the previous configuration. These methods can be classified according to the way in which the step direction and step size are chosen, see Figure (3.2):
(a) according to the energy $V(x)$, (b) according to the gradient of $V'(x)$, (c) according to the curvature of $V(x)$, (d) at random, and (e) according to a memory of the path followed so far. Energy minimisation can be based on only energy values and random steps (simplex methods), or on energy and energy gradient values (steepest-descent and conjugate-gradient methods), or on second-order derivatives of the energy (Hessian matrix methods). In MC methods the step direction is taken at random, and the step size is limited by the Boltzmann acceptance criterion: when the potential energy of the system changes by $\Delta V < 0$, the step in configuration space is accepted while for $\Delta V > 0$, the step is accepted with probability $\exp(-\Delta V/k_B T)$. In MD simulation the step is determined by the force, the negative of the local gradient $\partial V(x)/\partial x$, and by the inertia of the degrees of freedom, which serves as a short-time memory of the path followed so far. In SD simulation a random component is added to the force, the size of which is determined by the temperature of the system and the atomic masses and friction coefficients. In the potential-energy contour tracing (PECT) algorithm$^{49,50}$ and in the potential-energy annealing conformational search (PEACS) algorithm$^{51}$ the energy values are monitored and kept constant (PECT) or annealed (PEACS) in order to locate saddle points and pass over these. The catalytic tempering MC algorithm$^{52}$ is based on similar ideas. In MD and SD memory is built into the trajectory through inertial effects. Information on the history of the system can also be included in the force by averaging previous forces$^{53-57}$. There exists a large variety of search procedures based on stepping through configuration space using a combination of the five mentioned basic elements energy, gradient, Hessian, randomness and memory, combined in one way or the other$^{58}$.

The efficiency of search methods for biomolecular systems is severely restricted by the nature of the energy hypersurface $V(x)$ that is to be explored to find low energy regions. Due to the occurrence of a multitude of high energy barriers between local minima, the radius of convergence of the step methods is generally very small. Therefore, a variety of techniques have been developed to enhance the search and sampling power of searching methods. These are reviewed in Section 3.6.

### 3.5 Techniques to speed up a simulation

For a system containing $N$ atoms, the number of pairwise non-bonded interactions equals $N(N - 1)/2$. The computing time for the calculation of all these interactions is proportional to $N^2$. However, generally not all these interactions need to be computed. Different types of atomic interactions have different spatial ranges. The electrostatic interaction between two charges is proportional to $r^{-1}$, where $r$ is the distance between the charges. The dipolar interaction is of shorter range, that is, proportional to $r^{-3}$. The van der Waals interaction is of still shorter range, proportional to $r^{-6}$, only the first and second neighbour shells contribute significantly to the inter-
3.5. Techniques to speed up a simulation

Determinants of step direction and size

- Energy: \( V(x) \)
- Gradient: \( -\frac{\partial V}{\partial x} \)
- 2nd derivative: \( \frac{\partial^2 V}{\partial x \partial x'} \)
- Random:
- Memory:

<table>
<thead>
<tr>
<th>Method to generate configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM</td>
</tr>
<tr>
<td>MC</td>
</tr>
<tr>
<td>MD</td>
</tr>
<tr>
<td>SD</td>
</tr>
<tr>
<td>PECT, PEACS</td>
</tr>
</tbody>
</table>

Figure 3.2: Heuristic methods to search configuration space for configurations \( x \) with low energy \( V(x) \). EM: energy minimisation, MC: Monte Carlo, MD: molecular dynamics, SD: stochastic dynamics, PECT: potential energy contour tracing, PEACS: potential energy annealing conformational search.

action. In this situation the application of a cut-off radius beyond which no detailed atom-atom interactions are taken into account, but only represented in a mean-field, e.g. a reaction-field, sense, is appropriate. Once the nearest neighbours are found, which is an operation proportional to \( N^{59-61} \), the computation of the non-bonded interaction becomes proportional to \( N \) as well. Because electrostatic interactions are long-ranged, so-called particle-particle-particle-mesh techniques have been introduced\(^{61} \) to compute these efficiently\(^{62} \). The computational effort scales with \( N \log N \) due to the use of fast Fourier transform techniques\(^{63} \). An alternative is to approximate the medium beyond a given cut-off distance \( R_{rf} \) from a specific atom or molecule by a dielectric continuum of permittivity \( \varepsilon_{rf} \)\(^{64} \) and ionic strength \( I_{rf} \)\(^{65} \).

The length of the integration time step \( \Delta t \) is in MD or SD simulation limited by the highest frequency (\( \nu_{\text{max}} \)) motions occurring in the molecular system,

\[
\Delta t \ll \nu_{\text{max}}^{-1} = \tau. \tag{3.9}
\]

For a precise integration of the equations of motion, condition (3.9) must be satisfied. However, it may be that one is only interested in some average properties of the system. If those are not essentially dependent on ensemble fluctuations, the time step \( \Delta t \) may be lengthened beyond
condition (3.9).

In biomolecular systems three frequency ranges can be distinguished:

1. high-frequency bond-stretching forces $f^{hf}$ with an approximate oscillation or relaxation time $\tau^{hf}$ of about $10\,fs$,

2. low-frequency long-range Coulomb forces $f^{lf}$ with $\tau^{lf} \approx 1000\,fs$ or larger, and

3. the remaining intermediate-frequency forces $f^{if}$ with $\tau^{if} \approx 40\,fs$.

The contribution of these different forces to the atomic trajectories may be integrated using three different time steps, each satisfying condition (3.9) with the appropriate relaxation time $\tau$. When applied to bond-stretching forces, such a multiple-time-step (MTS) integration scheme saves a factor two to three in computing effort. When applied to the long-range Coulomb forces, the so-called twin-range method saves about a factor of five to ten due to the fact that the evaluation of these forces dominates the force calculation. The same kind of reasoning may also be applied to Monte Carlo simulations.

A few other numerical and conceptual tricks that may be used to speed-up a simulation are discussed elsewhere.

### 3.6 Search and sampling enhancement techniques

In Figure (3.3) three general types of search and sampling enhancement techniques are distinguished.

1. *Deformation or smoothening of the potential energy hypersurface in order to reduce barriers.*

   (a) Generally, a smoothening of the potential energy function $\mathcal{V}(\mathbf{x})$ allows for a faster search for its minima. This technique has been applied to different problems, such as structure determination based on X-ray diffraction or NMR spectroscopic data, conformational search and protein structure prediction. In method Ia of Figure (3.3) the electron density of a biomolecular crystal is smoothened by the omission of high-resolution diffraction intensities when backcalculating the electron density from these through Fourier transforms. This smoothening enhances the radius of convergence of the structure refinement.

   (b) When building protein structure from atom-atom distance data obtained from NMR, the convergence of the configurational search process is enhanced by gradually introducing distance restraints that connect atoms at longer distance along the polypeptide chain in the potential energy function. This is called a variable-target function method.
The hard core of atoms, *i.e.*, the strong repulsive interaction between atoms overlapping with each other, is responsible for many barriers on the energy hypersurface of a molecular system. These barriers can be removed by making the repulsive short-range interactions between atoms soft \(^7\). Soft-core atoms smoothen the energy surface and led to strongly enhanced sampling \(^8\).

In the diffusion-equation based deformation methods, \(^8\), the deformation of the energy surface driving a simulation is made proportional to the local curvature (second derivative) of the surface, which leads to a preferential smoothening of the sharpest peaks and valleys of the surface and very efficient search. The potential energy surface can be deformed in a great variety of ways. The corresponding search or sampling algorithms sail under an equally wide variety of names: potential-scaled MD \(^8\), stochastic tunneling \(^8\), q-jumping \(^8\), Nose-Hoover deformation \(^8\).

Incorporation of information on the energy hypersurface obtained during the search into the potential energy function is another possibility to enhance sampling. Once a local energy minimum is found, it is removed from the energy surface by a suitable local deformation of the potential energy function. This idea is the basis of the deflation method \(^8\) and the local-elevation search method \(^8\), which was recently also called meta-dynamics \(^8\). The method of conformational flooding \(^8\) is based on the same idea. Other variations can be found as well \(^8\).

Another way to introduce a memory into the search is the use of a potential energy term which is a running average over the atomic trajectories or ensemble generated so far, rather than its instantaneous value \(^5\). Application of this type of time-dependent or ensemble-dependent restraints in protein structure determination based on NMR or X-ray data leads to much enhanced sampling of the molecular configuration space \(^9\).

Barriers in the energy hypersurface can be circumvented by an extension of the dimensionality of the configuration space beyond the three Cartesian ones. The technique of energy embedding \(^9\) locates a low-energy conformation in a high-dimensional Cartesian space and gradually projects this conformation to three-dimensional Cartesian space while perturbing its energy and configuration as little as possible. Variations on the original procedure have been proposed \(^9\). Dynamic search methods can also be used in conjunction with an extension of the dimensionality. By performing MD in four-dimensional Cartesian space, energy barriers in three-dimensional space can be circumvented \(^\) and free energy changes calculated \(^\).

A long used standard technique to smoothen the energy surface is to freeze the highest-frequency degrees of freedom of a system through the application of constraints \(^\). Bond-length constraints are standardly applied in biomolecular simulation and allow for a four times longer time step size \(^\). High frequency
motion elimination can also be achieved through flexible constraints.

(i) A coarse-graining of the molecular model, which involves a reduction of the number of interaction sites, generally leads to a smoothening of the energy surface. This may allow the use of simulation time steps that are much (factor of 15) longer than the ones used in fine-grained (atomic) simulations.

2. Scaling of system parameters can also be used to enhance sampling.

(a) The technique of simulated temperature annealing involves simulation or search at a high temperature $T$ followed by gradual cooling. By raising the temperature, the system may more easily surmount energy barriers, so a larger part of configurational space can be searched. The technique of simulated temperature annealing has been widely used in combination with MC, MD and SD simulation. An example of potential energy annealing can be found in. The so-called J-walking algorithm also uses temperature variations to enhance the sampling.

(b) One way of keeping a constant temperature in a simulation is to use an additional equation that linearly couples the actual temperature $T(t)$ to a reference or heat-bath temperature $T_{ref}$:

$$
\frac{dT(t)}{dt} = \tau_T^{-1} (T_{ref} - T(t))
$$

The coupling strength being determined by the coupling time $\tau_T$. When choosing this parameter close to the time step ($\tau_T \geq \Delta t$), the kinetic energy or velocities are enhanced when the system’s potential energy increases and the velocities are reduced in low potential energy regions. This enhances the sampling.

(c) Scaling of atomic masses can be used to enhance sampling. In the classical partition function and in case no constraints are applied, the integration over the atomic momenta can be carried out analytically, separately from the integration over the coordinates. Thus, the atomic masses do not appear in the configurational integral, which means that the equilibrium (excess) properties of the system are independent of the atomic masses. This freedom can be exploited in different ways to enhance the sampling. By increasing the mass of specific parts of a molecule, their relative inertia is enhanced, which eases the surmounting of energy barriers, and may allow for longer time steps. A reduction of the mass of the solvent molecules has been shown to lead to enhanced sampling of the folding/unfolding equilibrium of a polypeptide in explicit solvent simulation. The canonical adiabatically free energy sampling (CAFES) algorithm also exploits inertia to speed up the occurrence of rare events.

(d) Enhanced sampling by a mean-field approximation is obtained by separating the biomolecular system into two parts, A and B, each of which moves in the average
3.6. Search and sampling enhancement techniques

field of the other. The initial configuration of the system consists of $N_A$ identical copies of part $A$ and $N_B$ identical copies of part $B$, where the positions of corresponding atoms in the identical copies may be chosen to be identical. The force on atoms in each copy of part $A$ exerted by the atoms in all copies of part $B$ is scaled by a factor $N_B^{-1}$, in order to obtain the mean force exerted by part $B$ on the individual atoms of part $A$. The force on atoms in each copy of part $B$ exerted by the atoms in all copies of part $A$ is scaled by a factor $N_A^{-1}$, in order to obtain the mean force exerted by part $A$ on the individual atoms of part $B$. The forces between the different copies of part $A$ are zero, and so are the forces between the different copies of part $B$. The MD simulation involves the integration of Newton’s equation of motion, $f = ma$, for all copies of parts $A$ and $B$ simultaneously. Thus one obtains $N_A$ individual trajectories of part $A$ in the mean field of part $B$ and vice versa. This comes at the loss of correct dynamics: Newton’s third law, $f_{AB} = -f_{BA}$ is violated. The technique only enhances efficiency when the system is partitioned into parts of very different sizes, e.g. $\text{size}(A) << \text{size}(B)$ and the bigger part is represented by one copy: $N_B = 1$. Locally enhanced searching and sampling (LES) procedures based on a mean-field approximation have been proposed in different forms\textsuperscript{112-117}.

3. Multi-copy simulation with a given relation between the copies can also be used to enhance searching and sampling.

In the mean-field approach sketched before multiple copies of a part of the system were simulated. This idea has been used in different ways to enhance searching and sampling, see Figure (3.3).

(a) In genetic algorithms\textsuperscript{118} a pool of copies of the biomolecular system in different configurations is considered and new configurations are created and existing ones deleted by mutating and combining (parts of) configurations according to a given set of rules.

(b) In the so-called replica-exchange algorithm multiple copies of the system are simulated by MC, MD or SD, each at a distinct temperature. From time to time copies at adjacent temperatures are exchanged using an exchange probability based on the Boltzmann factor (3.6). This leads in the limit of infinite sampling to Boltzmann-distributed (canonical) ensembles for each temperature\textsuperscript{119}. So-called multi-canonical algorithms are a generalisation of this procedure\textsuperscript{120}. This type of algorithm has been used to simulate proteins in vacuo\textsuperscript{119}. The inclusion of solvent degrees of freedom may impair the efficiency of the algorithm\textsuperscript{121}. Dynamical information is lost in the exchanges. A variety of schemes of this type has been recently proposed: generalised-ensemble algorithms\textsuperscript{120,122}, local and partial replica-exchange\textsuperscript{123}, parallel replica method\textsuperscript{124}, combinations of parallel tempering, multi-canonical and multi-
ple histogram methods\textsuperscript{125}, and broad-histogram MC\textsuperscript{126,127}.

(c) The so-called SWARM type of MD\textsuperscript{128} is based on the idea of combining a collection or swarm of copies of the system each with its own trajectory into a cooperative multi-copy system that searches configurational space. To build such a cooperative multi-copy system, each copy is, in addition to physical forces due to $\psi(x)$, subject to (artificial) forces that drive the trajectory of each copy toward an average of the trajectories of the swarm of copies, in analogy to the fact that intelligent and efficient behaviour of a whole swarm of insects can be achieved even in the absence of any particular intelligence or forethought of the individuals. SWARM-MD is less attracted by local minima and is more likely to follow an overall energy gradient toward the global energy minimum. Other multi-copy methods can be found\textsuperscript{129–131}.

Techniques to enhance the searching and sampling power of simulation methods

\begin{enumerate}
\item \textbf{Deformation or smoothening of the potential energy surface}
\begin{enumerate}
\item omission of high-resolution structure factor data in structure refinement based on X-ray diffraction data
\item gradual introduction of longer-range distance restraints in variable target structure refinement based on NMR NOE data
\item softening of the hard core of atoms in the non-bonded interaction (soft-core atoms)
\item reduction of the ruggedness of the energy surface through a diffusion-equation type of scaling
\item avoiding the repeated sampling of an energy well through local potential energy elevation or conformational flooding
\item softening of geometric restraints derived from experimental (NMR, X-ray) data through time-averaging of these
\item circumvention of energy barriers through an extension of the dimensionality of the Cartesian space (4D-MD)
\item freezing of high-frequency degrees of freedom through the use of constraints
\item coarse-graining the model by reduction of the number of interaction sites
\end{enumerate}
\item \textbf{Scaling of system parameters}
\begin{enumerate}
\item temperature annealing
\item tight coupling to heat bath
\item mass scaling
\item mean-field approaches
\end{enumerate}
\item \textbf{Multi-copy searching and sampling}
\begin{enumerate}
\item genetic algorithms
\item replica-exchange and multi-canonical algorithms
\item cooperative search: SWARM
\end{enumerate}
\end{enumerate}

Figure 3.3: \textit{Techniques to enhance the searching and sampling power of simulation methods. For details see text of Section 3.6.}

The overviews of Figures 3.2 and 3.3 are meant to offer a hand when choosing a combination of search or sampling methods with various enhancement techniques that will be appropriate to model the particular system and energy function of interest, leading to an efficient calculation of the requested properties.
3.7 Biasing the search, sampling or simulation

The search or sampling enhancement methods discussed in the previous section did so without a particular bias being imposed on the molecular system. However, if the particular barriers of the potential energy function or hypersurface that block access to low-energy parts of the surface can be identified, this knowledge can be put into the form of a biasing potential energy term to be added to the Hamiltonian, which will guide the trajectory in a required direction. A variety of such biased searching or sampling methods exists.

Since high-frequency motions are generally not of great interest, one may bias an MD simulation in the direction of slower modes by filtering out the high frequencies from the spectrum during a simulation. This approach has been used in various studies. Another possibility is to bias the motion in an MD simulation in the direction of the principal components as obtained from the trajectory so far. This bias should enhance the exploration of larger amplitude modes of the molecular system. Yet another method couples the collective modes of a system to a bath of higher temperature than the other modes. This enhances the sampling along the collective modes.

Recently, a method to enhance sampling of rare events was proposed, which makes use of distance or torsional-angle restraints to overcome an energy barrier separating two metastable states, or to stabilise a transition state between the two metastable states. The latter states are not subject to restraints, which allows one to determine the free energy difference between the two metastable states without the need to choose a physically realistic pathway connecting them.

3.8 Sampling or simulation along pathways

Dynamical processes in biomolecular systems may occur on time scales far beyond the ones that are accessible through standard MD simulations. If these processes are intrinsically slow, i.e. require an extensive sampling of configuration space, not much can be done to speed up their simulation without destroying the dynamics of the system. If, however, these processes are rare, i.e. they do not occur often, but when occurring they are fast, there are possibilities to enhance the sampling of these rare processes. Generally they are characterised by the need to pass over a high-energy barrier separating two meta-stable states.

The oldest approach to sample transitions is to define a reaction coordinate or transition pathway and to sample along this path using a biasing potential energy term and umbrella sampling. A variation using MD is so-called targeted MD. A more sophisticated methodology is transition path sampling, which finds transition pathways for infrequent events and requires no knowledge of the transition mechanism or transition state, only the end states need be defined. Although this method is more powerful than traditional reaction-coordinate sampling, the requirement of a proper definition of the two end states restricts its applicability.
The method and its applications have been recently reviewed\textsuperscript{145,146}. A novel extension is called transition interface sampling\textsuperscript{147}.

Other methods to determine and sample transition pathways are the finite temperature string method, which generates a tube in configuration space between the end states, inside which conformational changes occur with high probability. This leads to an increased rate of occurrence of the rare transitions\textsuperscript{148,149}. A minimum-energy path of a transition can be obtained through the nudged elastic band method\textsuperscript{150-153}.

An alternative way to speed up rare events was called “hyper-MD”\textsuperscript{154,155}. It uses a biasing potential to guide the dynamics away from the end states. Yet another scheme is called coarse MD\textsuperscript{156}.

### 3.9 Use of other than spatial molecular coordinates when searching or sampling

The methodology discussed in the previous sections is based on the use of molecular spatial coordinates as variables which are sampled. This approach is widely used, but may not lead to effective solutions when the energy hypersurface is characterised by extremely high potential energy barriers separating different tightly packed, low energy conformations. Problems of this type are the docking of inhibitor or substrate molecules into an active site of an enzyme or the prediction of dominant side-chain conformations of amino acid residues in mutated proteins. For such cases one may use a rather different search and sampling technique, in which not only the molecular coordinates $x$ serve as variables, but also the Boltzmann probability $P_\alpha$ of occurrence of a molecular conformation $x_\alpha$.

\begin{equation}
P_\alpha = \frac{e^{-\mathcal{V}(x_\alpha)/k_B T}}{\sum_{\alpha'} e^{-\mathcal{V}(x_{\alpha'})/k_B T}} \tag{3.11}
\end{equation}

The computational problem is now to minimise the average potential energy

\begin{equation}
\langle E \rangle = \sum_\alpha P_\alpha \mathcal{V}(x_\alpha) \tag{3.12}
\end{equation}

subject to condition (3.11). That is, one wishes to find a Boltzmann distributed ensemble of configurations for a very high-dimensional and complex interaction function $\mathcal{V}(x)$. Since the average energy $\langle E \rangle$ in (3.12) depends on both, conformational coordinates $x_\alpha$ and conformational probabilities $P_\alpha$, four types of search or optimisation algorithms to obtain a set of $(x_\alpha, P_\alpha)$ values that represent a Boltzmann ensemble may be distinguished\textsuperscript{58}.

1. Conformational coordinates $x_\alpha$ are treated as variables, probabilities $P_\alpha$ implicitly satisfy (3.11). This is the classical conformational search problem as discussed in the previous
sections, in which the molecular coordinates $x_\alpha$ are changed according to classical (constant $T$) mechanics (MD) or using a Markov probability chain (MC) such that the probabilities $P_\alpha$ automatically satisfy (3.11).

2. Multiple conformations $x_\alpha$ are used simultaneously, but kept fixed ($\alpha_1, \alpha_2, \alpha_3, \ldots$), probabilities $P_\alpha$ are treated as variables, which follow from (3.11). This approach works when the relevant conformations $x_{\alpha_1}, x_{\alpha_2}, \ldots$ can be easily identified a priori\textsuperscript{157,158}.

3. Multiple conformations $x_\alpha$ are used simultaneously as variables that change according to classical equations of motion, probabilities $P_\alpha$ are treated as parameters that adiabatically follow the variation of $x_\alpha$ according to (3.11). This approach\textsuperscript{58} has been demonstrated using a cyclic peptide\textsuperscript{158}.

4. Multiple conformations $x_\alpha$ and probabilities $P_\alpha$ are used simultaneously as variables that change according to classical equations of motion. This is a generalisation of the previous approach\textsuperscript{158}. The Boltzmann relation (3.11) can be imposed on the variables $(x_\alpha, P_\alpha)$ either in the form of a penalty function for $P_\alpha$ which is added to the standard interaction function $V(x_\alpha)$, or in the form of a constraint to $P_\alpha$, which is to be satisfied when the equations of motion for $(x_\alpha, P_\alpha)$ are integrated.

### 3.10 Discussion

An overview of the types of methods that are currently used in biomolecular modeling to search or sample or dynamically move through the configurational space of a molecular system was given. Since in general the configurational space is too large to be completely searched or sampled, the various methods (Section 3.4) and techniques aim at reducing the size of the problem (Section 3.3), or at using particular algorithms to enhance efficiency (Section 3.5), or at transforming the problem into a more tractable one for which solutions can be found that are good approximations to solutions of the original problem (Section 3.6). Generally, saving computational effort by the techniques presented has its price: the accuracy of the generated ensemble or trajectory is decreased depending on the search or sampling enhancement technique used or depending on the type of degrees of freedom that are omitted from the calculation. Whether such a loss in accuracy of particular properties is acceptable depends on the goals of a modeling study. In this respect one may distinguish four different degrees of distortion of the correct result induced by search or sampling (enhancement) techniques.

1. Techniques that preserve the correct dynamics of the system (no distortion).

2. Techniques that distort the dynamics, but generate a correct Boltzmann ensemble.
3. Techniques that distort dynamics and ensemble; they only generate an arbitrary collection of molecular configurations.

4. Techniques that yield neither dynamics nor an ensemble or arbitrary set of molecular configurations, but only one molecular configuration.

For example, when relative free energies of binding or complexation are to be obtained, correct dynamics is not required, only a proper ensemble\textsuperscript{159}.

If the biomolecular modeling problem can be formulated in terms of particular conformational states, the search and sampling problem is reduced to pathways connecting such (end) states (Section 3.8) and efficiency may be enhanced using biasing techniques (Section 3.7). For example, in free energy calculations unphysical pathways may be used to obtain the relative free energies of two end states\textsuperscript{159}.

The bulk of the methods and techniques that were discussed are based upon variation of spatial molecular coordinates. Yet methods that use other molecular coordinates have been proposed and found some use (Section 3.9).

Of the many search and sampling methods and enhancement techniques reviewed a few are very effective: use of soft-core atoms, local-elevation simulation and its derivatives, replica-exchange simulation and generalised-ensemble methods. When end states are known, transition path sampling is a powerful method. For other reviews of search and sampling methodology we refer to\textsuperscript{26,58,74,145,160–162}. The present one is meant to support the practical biomolecular modeler when choosing a combination of methods and tricks that will be particularly suited to the specific problem, \textit{i.e.} molecular system and properties to be computed, of interest.

### 3.11 Acknowledgements

Financial support by the National Center of Competence in Research (NCCR) Structural Biology of the Swiss National Science Foundation (SNSF) is gratefully acknowledged.

### 3.12 Bibliography


3.12. Bibliography


Chapter 3. searching methods reviewed


3.12. Bibliography


Chapter 4

Investigation of sampling efficiency using configurational entropy as a measure

4.1 Summary

Configurational entropy calculations were used to measure sampling of phase space of butane in vacuo. Three different simulation techniques were compared. Monte-Carlo simulations were used as reference and compared with stochastic dynamics simulations and replica-exchange stochastic dynamics simulations. Two temperature regimes were investigated. Temperatures above 250 K where the calculated configurational entropies converged for all simulation techniques, and temperatures below 200 K, where only the configurational entropies from the Monte-Carlo simulations were converged. Using the replica-exchange method sampling efficiency was only slightly improved. This was found to be due to a separation of the replicas in two separated sets with only little exchange between those. Increasing the switching frequency from 1 ps$^{-1}$ to 100 ps$^{-1}$ led to marginally better sampling efficiency. The configurational entropy was calculated using the Schütter method. For comparison, also the configurational entropy of the dihedral angle was calculated from its probability distribution using Shannon’s formula. At low temperatures, the results obtained from these different calculation methods seemed to diverge. Using Monte-Carlo simulations in internal coordinates, it could be shown that rotational fitting of the structures had a significant impact on the configurational entropy calculated by the Schlitter method. Without applying rotational fitting, Schlitter’s configurational entropy corresponds closely to Shannon’s entropy calculated from the dihedral-angle probability distribution.
4.2 Introduction

Computer simulation of biomolecular systems is often limited by the short time-scale that is achievable, even if using modern computer hardware. Parallelisation of calculations substantially increases this time-scale. Nevertheless, maintaining a good scaling up to a large number of parallel processes requires a highly optimized computational environment and software. Replica-exchange simulation, which enhances sampling of conformational space by simultaneously simulating a number of independent replicas of a biomolecular system, which only occasionally letting them interact, scales very well with increasing computational resources while not requiring extremely efficient communication between the single computational nodes.1-10

In this article configurational entropy is used as a measure of phase-space sampling. First, simulations at high temperatures were used to confirm that in the event of complete sampling, configurational entropies calculated from replica-exchange simulations were identical to the ones calculated using independent simulations. Then, the simulations were repeated at much lower temperatures and convergence behaviour of their entropy was investigated. Also, the effect upon sampling efficiency of altering the exchange frequencies used for the replica-exchange simulations was tested. The calculation of configurational entropies at low temperatures is shortly investigated as well.

4.3 Method

In a replica-exchange simulation, a number of non-interacting replicas are simulated simultaneously at different conditions (e.g. at different temperatures). After a given simulation time, an exchange between two replicas is attempted, followed by another (individual) simulation period. The probability of the global state $S'$ consisting of $N'$ replicas is proportional to its weight factor

$$W(S') = \exp \left( - \sum_{s=1}^{N'} \beta_s H(r, p) \right),$$

with $\beta_s = 1/k_B T_s$, $k_B$ is Boltzmann's constant and $T_s$ the temperature of replica $s$. Here, the notation $r \equiv (r_1, r_2, ..., r_N)$ to indicate a $N$-particle configuration with position $r_i$ of particle $i$, and analogous for the momenta $p$ is used.

After a fixed number of MD integration steps, a MC exchange between two replicas is attempted (changing from state $S'$ to state $S''$). To sample canonical ensembles at each temperature, the detailed balance condition on the transition probability $w(S' \rightarrow S'')$

$$W(S') w(S' \rightarrow S'') = W(S'') w(S'' \rightarrow S')$$

has to be fulfilled. This can be satisfied, for instance, by the usual Metropolis criterion for the
4.3. Method

probability \( p(s' \leftrightarrow s'') \) of an exchange of the two replicas \( s' \) and \( s'' \),

\[
p(s' \leftrightarrow s'') = \frac{W(S'')}{W(S')} = \frac{w(s' \rightarrow s'')}{w(S'' \rightarrow S')} = \begin{cases} 
 1 & \text{for } \Delta \leq 0, \\
 \exp(-\Delta) & \text{otherwise},
\end{cases}
\]

with

\[
\Delta = (\beta s' - \beta s'')(U(r_{s'}) - U(r_{s})),
\]

where \( U(r) \) is the potential energy associated with the configuration \( r \). If the exchange was successful, the momenta of the exchanged replicas are scaled to correspond to their new temperatures.

Replica-exchange simulation is used to improve sampling of configurational space of the lower temperature replicas. While in replica-exchange simulations Newtonian dynamics is clearly violated at the times of replica exchange, thermodynamic averages do correspond to canonical ensemble averages. Thermodynamic quantities are determined by energy and entropy of the system. The energy is directly taken into account by the Metropolis criterion (Equation 4.3), whereas the entropy is accounted for by the dynamics of the system. As we have just observed, the dynamics of a replica-exchange simulation do not coincide with those of truly independent simulations. Therefore, the effects of using the replica-exchange method on the entropy of the system at the different temperatures were investigated. To calculate the configurational entropy Schlitter's method was used\(^{12}\).

\[
S_{conf} = \frac{k_B}{2} \ln \left( \det \left( \mathbf{1} + \frac{e^2}{\beta h^2} \mathbf{D}_r \right) \right),
\]

where \( e \) is Euler's number, \( \mathbf{1} \) the unit matrix, \( \mathbf{D}_r \) is the mass weighted covariance matrix

\[
\mathbf{D}_r = \mathbf{M}^{1/2} \mathbf{C}_r \mathbf{M}^{1/2}
\]

with the mass matrix \( \mathbf{M} \), the covariance matrix \( \mathbf{C}_r \)

\[
\mathbf{C}_r = \langle (\mathbf{r} - \langle \mathbf{r} \rangle) \otimes (\mathbf{r} - \langle \mathbf{r} \rangle) \rangle
\]

and with the index \( r \) indicating the use of a Cartesian coordinate system. Angular brackets \( \langle Q \rangle \) denote the time average of a quantity \( Q \). As a comparison, also configurational entropies from probability distributions of a selected degree of freedom were calculated using Shannon's formula\(^{13}\)

\[
S_{sh} = -k_B \int_{-\infty}^{\infty} P(q) \ln(P(q)) dq,
\]

where \( P(q) \) is the probability density of a degree of freedom along its coordinate \( q \).
4.4 Model

To show that configurational entropies calculated from independent and from replica-exchange stochastic-dynamics simulations are identical in the event of complete sampling of configurational space, a very simple test system, where complete sampling can be reached easily, was selected. We chose butane in vacuo. In order to have reasonable exchange probabilities with temperature differences of 10K or 20K between the replicas 200 non-interacting butanes were simulated together. Non-interacting means that standard GROMOS 45A3 force field terms were used for the intra-molecular interactions whereas no inter-molecular interactions were calculated. Also the average temperature controlled through coupling to the stochastic temperature bath is closer to the specified value due to averaging over more degrees of freedom. All simulations were started from a structure where all 200 butanes are in the trans-conformation. The stochastic-dynamics friction coefficients were uniformly set to 6.658 ps^-1 for all atoms, conforming to the diffusion constant obtained from liquid-phase molecular dynamics simulations. The total simulation time was 1 ns, which corresponds to 1'000'000 steps of 1 fs length. In the independent simulations, configurations were saved every 1 ps for later analysis, in replica-exchange simulations this was done every 0.2 ps. Monte-Carlo simulations were carried out for a single butane and only for the dihedral-angle degree of freedom, with 50'000 moves per simulation. The moves involved changing the dihedral angle by a random amount uniformly distributed in the range from −60 to 60 degrees and were accepted or rejected according to the Metropolis criterion. All configurations were saved in Cartesian coordinates for later analysis.

4.5 Results

In Table 4.1 potential energies, configurational entropies calculated using Equation 4.5 ($S_{conf}$) and the dihedral-angle configurational entropy (using Equation 4.8, $S_{dih}^{th}$) are shown for temperatures between 200K and 400K. The convergence behaviour of the configurational entropy can be seen in Figure 4.1.

One can distinguish three different types of behaviour. The first type applies to temperatures above 200K, where the configurational entropy obtained from 1 ns of stochastic-dynamics simulation was converged. The second is exhibited by intermediate temperatures, from 160K to 200K, and corresponds to non-converged configurational entropies. Extending the simulation time by a factor of two to three would probably be enough to get the entropies calculated from these simulations converged as well. Interestingly enough, for the lowest temperature simulations, the configurational entropies were again converged, though to significantly lower values. This was due to trapping of the simulation in the local (and in this case also global) minimum on the (free) energy hypersurface, where the starting configuration was located. This is shown in Figure 4.2, where a time-series of the dihedral-angle distribution is depicted. In the low
4.5. Results

<table>
<thead>
<tr>
<th>$T$</th>
<th>$E_{\text{pot}}$</th>
<th>$S_{\text{conf}}$</th>
<th>$S_{\text{dih}}^s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.0</td>
<td>8.4</td>
<td>29.0</td>
</tr>
<tr>
<td>130</td>
<td>1.9</td>
<td>13.6</td>
<td>32.2</td>
</tr>
<tr>
<td>160</td>
<td>3.0</td>
<td>26.6</td>
<td>36.5</td>
</tr>
<tr>
<td>190</td>
<td>3.9</td>
<td>39.4</td>
<td>38.3</td>
</tr>
<tr>
<td>220</td>
<td>4.8</td>
<td>45.4</td>
<td>39.6</td>
</tr>
<tr>
<td>250</td>
<td>5.6</td>
<td>51.5</td>
<td>40.6</td>
</tr>
<tr>
<td>280</td>
<td>6.5</td>
<td>55.5</td>
<td>41.4</td>
</tr>
<tr>
<td>340</td>
<td>8.1</td>
<td>60.6</td>
<td>42.7</td>
</tr>
<tr>
<td>400</td>
<td>9.7</td>
<td>65.6</td>
<td>43.7</td>
</tr>
</tbody>
</table>

Table 4.1: Potential energy ($E_{\text{pot}}$ in kJ mol$^{-1}$), total configurational entropy ($S_{\text{conf}}$ in JK$^{-1}$ mol$^{-1}$) and configurational entropy of the dihedral angle distribution ($S_{\text{dih}}^s$ in JK$^{-1}$ mol$^{-1}$) of the independent (1 ns) simulations at temperatures between 100 K and 400 K.

temperature simulations, almost no butane molecules did the transition from trans- to gauche-conformation.

The configurational entropies $S_{\text{conf}}$ calculated for the butane molecules showed a much stronger dependence on temperature than the entropy $S_{\text{dih}}^s$ calculated from the dihedral-angle distributions. Indeed $S_{\text{conf}}$ seems to be too low at temperatures under 200 K. Using the expression for the entropy of a Gaussian probability distribution function

$$S_{\text{gauss}} = -kB \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \ln \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} dx,$$

the width of this Gaussian distribution corresponding to a given entropy $S_{\text{gauss}}$ can be calculated. Figure 4.3 shows the Gaussian distributions corresponding to the total configurational entropies $S_{\text{conf}}$ calculated using Equation 4.5 and the dihedral-angle probability distributions (with their entropies listed in Table 4.1). They are quite different. To further investigate this troubling discrepancy, Monte-Carlo simulations of a single butane at low temperature were performed. These Monte-Carlo simulations were done in internal coordinates, with the dihedral angle as the only (non-rigid) degree of freedom. A comparison of configurational entropies calculated from these Monte-Carlo simulations at different temperatures using Equation 4.8 and 4.5 with and without rotational fitting of the butane molecule to the initial structure is given in Table 4.2. As the Monte-Carlo simulations are carried out in internal coordinates, the first three atoms of the butane molecule were fixed, only the fourth one was changing. This exactly meets the condition of “anchored Cartesian” coordinates$^{17}$ and is distinct from rotational fitting$^{18}$. It seems that rotational fitting, at low temperatures, leads to overestimation of the rotational entropy and underestimation of the configurational entropy.
Figure 4.1: Configurational entropy ($S_{\text{conf}}$) time-series of independent 1 ns stochastic dynamics simulations of (non-interacting) butane at temperatures between 100 K and 400 K.

As a first test, a replica-exchange stochastic-dynamics simulation with seven replicas evenly spaced at temperatures from 280 K to 400 K was used to calculate the same properties (results are shown in Table 4.3). The exchange probabilities between pairs of replicas during the simulation were between 0.22 and 0.39 and the overlap of the potential energy distributions for the different temperatures is shown in Figure 4.4. All calculated properties corresponded very well to the ones calculated from independent simulations and varied only very little for the three different replica-exchange frequencies of $100 \text{ ps}^{-1}$, $1 \text{ ps}^{-1}$ and $0.01 \text{ ps}^{-1}$ that were used.

Next, a closer look was taken at a replica-exchange stochastic-dynamics simulation using eleven replicas at temperatures evenly spaced from 100 K to 200 K. This case should correspond more closely to applications of more complex systems where incomplete sampling or configurational trapping is much more likely to occur. For a replica-exchange frequency of $1 \text{ ps}^{-1}$, the time-series of replica temperatures is shown in Figure 4.5. It can be clearly seen that the replicas at 140 K and above frequently exchanged, and to a lesser extent also the ones below 140 K. But in between these two sets of replicas there were hardly any exchanges observed.
4.5. Results

![Figure 4.2](image)  

**Figure 4.2**: *Dihedral-angle distributions of independent 1 ns stochastic dynamics simulations of (non-interacting) butane at temperatures of 100K, 160K, 190K, 220K, 250K, and 400K. Distributions after the first 100 ps shown in black, then cumulative distributions all 100 ps in gray, the last one in red.*

at all. This significantly reduced any enhanced sampling one might expect from the replica exchange simulations as even for the replica at 130K the butanes seem to be trapped in the trans-conformation (see Figure 4.1). The bigger separation between the replica at 130K and the one at 140K is visible in Figure 4.6 showing the overlap of the potential energy distributions for the various replicas. Transition probabilities for this replica-exchange simulation and a second one using an exchange frequency of 100 ps$^{-1}$ are shown in Table 4.4. Interestingly, this reduction in exchange probability between 130K and 140K is less pronounced for the simulation with much higher exchange frequency. The exchange probabilities are still fairly low, but because of the many more exchanges that were tried, still enough of them succeeded to improve the sampling. In Figure 4.7 the time-series of the configurational entropy for some of the replicas is plotted. Comparing the entropies to the ones calculated from the independent simulations (dashed lines), the replica at 100K showed considerably better sampling. The dihedral-angle entropy calculated (see Table 4.5) was quite close to the converged one calculated from the
Table 4.2: Configurational entropy (in $\text{JK}^{-1}\text{mol}^{-1}$) of a Monte-Carlo simulation of a single butane in vacuo using 50000 moves calculated using the Shannon formula (Equation 4.8) and the probability density ($S_{\text{sh}}^{\text{dih}}$) or using a trajectory of Cartesian coordinates and the Schlitter-method (Equation 4.5) with rotational fitting ($S_{\text{conf}}$) or without fitting ($S_{\text{conf}}(\text{no fit})$). During the simulation (and in the trajectory), the first three atoms do not move and the dihedral angle is the only changing degree of freedom throughout the simulation.

<table>
<thead>
<tr>
<th>$T$</th>
<th>$S_{\text{dih}}^{\text{sh}}$</th>
<th>$S_{\text{conf}}$</th>
<th>$S_{\text{conf}}(\text{no fit})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>34.1</td>
<td>17.4</td>
<td>31.9</td>
</tr>
<tr>
<td>120</td>
<td>36.3</td>
<td>23.0</td>
<td>37.3</td>
</tr>
<tr>
<td>140</td>
<td>38.1</td>
<td>26.3</td>
<td>40.8</td>
</tr>
<tr>
<td>160</td>
<td>39.2</td>
<td>28.2</td>
<td>42.8</td>
</tr>
<tr>
<td>180</td>
<td>40.3</td>
<td>30.1</td>
<td>44.7</td>
</tr>
<tr>
<td>200</td>
<td>41.0</td>
<td>31.4</td>
<td>46.1</td>
</tr>
</tbody>
</table>

Table 4.3: Potential energy ($E_{\text{pot}}$ in $\text{kJ}\text{mol}^{-1}$), total configurational entropy ($S_{\text{conf}}$ in $\text{JK}^{-1}\text{mol}^{-1}$) and configurational entropy of the dihedral angle distribution ($S_{\text{dih}}$ in $\text{JK}^{-1}\text{mol}^{-1}$) of replica-exchange simulations using seven replicas evenly spaced in temperatures between 280K and 400K and the indicated exchange frequencies (100 ps$^{-1}$, 1 ps$^{-1}$, and 0.1 ps$^{-1}$).

<table>
<thead>
<tr>
<th>$T$</th>
<th>100 ps$^{-1}$</th>
<th>1 ps$^{-1}$</th>
<th>0.1 ps$^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_{\text{pot}}$</td>
<td>$S_{\text{conf}}$</td>
<td>$S_{\text{dih}}$</td>
</tr>
<tr>
<td>280</td>
<td>6.4</td>
<td>54.6</td>
<td>41.4</td>
</tr>
<tr>
<td>340</td>
<td>8.1</td>
<td>60.3</td>
<td>42.7</td>
</tr>
<tr>
<td>400</td>
<td>9.7</td>
<td>65.3</td>
<td>43.7</td>
</tr>
</tbody>
</table>

Table 4.4: Exchange probability ($p$) of neighbouring pairs of a replica-exchange simulation with an exchange frequency of 100 ps$^{-1}$, 1 ps$^{-1}$ and 10 ps$^{-1}$ of (non-interacting) butane using eleven replicas at temperatures evenly spaced between 100K and 200K.

<table>
<thead>
<tr>
<th>freq</th>
<th>100</th>
<th>110</th>
<th>120</th>
<th>130</th>
<th>140</th>
<th>150</th>
<th>16</th>
<th>170</th>
<th>180</th>
<th>190</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>0.06</td>
<td>0.09</td>
<td>0.09</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
<td>0.21</td>
<td>0.26</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.09</td>
<td>0.08</td>
<td>0.16</td>
<td>0.09</td>
<td>0.20</td>
<td>0.22</td>
<td>0.18</td>
<td>0.25</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.08</td>
<td>0.10</td>
<td>0.19</td>
<td>0.09</td>
<td>0.13</td>
<td>0.17</td>
<td>0.34</td>
<td>0.29</td>
<td>0.33</td>
</tr>
</tbody>
</table>
4.5. Results

Figure 4.3: Dihedral angle distribution of the (1 ns) independent stochastic-dynamics simulations at temperatures between 100 K and 200 K (black lines) and Gaussian distributions corresponding to the entropies calculated for these simulations using the Schütter method.

Monte-Carlo simulation (see Table 4.2). The biggest difference is visible at 160 K where an almost converged entropy is obtained compared to the much lower and still rising one from the independent simulation. Interestingly, even the simulation at the highest temperature, 200 K, profited from replica-exchange and showed an entropy representing increased sampling. The replica-exchange simulation using an exchange frequency of 1 ps\(^{-1}\) showed the same improvements for 160 K and 200 K, but almost no improvement for 100 K. An additional measure for the conformational sampling is provided by the width of the dihedral-angle distribution. The time-series of this width for the replica-exchange simulation using the high exchange frequency is shown in Figure 4.8. During intervals with frequent exchanges between two replicas the changes between two consecutive points can be seen as “line splitting”. Again, increased sampling at 100 K is visible, whereas for the replica at 160 K it is not evident in this plot.
Figure 4.4: Potential energy distributions showing the overlap between the replicas in a stochastic-dynamics replica-exchange simulation of (non-interacting) butane at seven temperatures evenly spaced between 280 K and 400 K and an exchange frequency of 1 ps⁻¹.

Table 4.5: Potential energy (\(E_{\text{pot}}\) in kJ/mol⁻¹), total configurational entropy (\(S_{\text{conf}}\) in JK⁻¹mol⁻¹) and configurational entropy of the dihedral angle distribution (\(S_{\text{sh}}^{\text{dih}}\) in JK⁻¹mol⁻¹) of replica-exchange simulations using seven replicas evenly spaced in temperatures between 100 K and 200 K and the indicated exchange frequencies (100 ps⁻¹, 1 ps⁻¹, and 0.1 ps⁻¹).
4.6 Discussion

In this work, efficiency of sampling of configurational phase-space for a simple system was investigated and compared for independent and replica-exchange stochastic-dynamics simulations. For high temperatures, when all properties calculated were converged, both techniques result in identical thermodynamic averages. At low temperatures, replica-exchange simulation is at an advantage, although sufficiently high exchange-rates need to be achieved. Increased exchange rates can be obtained by increasing the frequency of exchange trials. Even a quite fast exchange frequency did not show any adverse effects on the calculated thermodynamic properties.

Rotational fitting is crucial to separate configurational entropy from rotational (and translational) entropy. It seems, that at least for low temperatures, the procedure by which the rotational fit is carried out has a significant impact on this separation.

Figure 4.5: Time-series of the temperature the replicas are at during a replica-exchange stochastic-dynamics simulation using an exchange frequency of $1 \text{ps}^{-1}$ and eleven replicas spaced evenly between 100K and 200K. Two loosely coupled regimes are identified by representing replicas mostly belonging to the regime with lower temperatures by dashed lines and the ones mostly belonging to the higher temperature regime by solid lines.
Figure 4.6: Potential energy distributions showing the overlap between the replicas in a (1 ns) stochastic-dynamics replica-exchange simulation of (non-interacting) butane at eleven temperatures evenly spaced between 100 K and 200 K and an exchange frequency of 1 ps$^{-1}$. The two regimes separated by a small exchange probability are shown using dashed lines for ensembles belonging to the lower temperature one and solid lines for the others.

For real replica-exchange simulations careful analysis of the exchange probabilities and of separation of replicas into less exchanging subsets is necessary to avoid being caught unaware in a local-minimum trap. Nevertheless, replica-exchange simulation showed significantly improved sampling of conformational space for the simple test system at low temperatures.

4.7 Acknowledgements

Financial support by the National Center of Competence in Research (NCCR) Structural Biology of the Swiss National Science Foundation (SNSF) is gratefully acknowledged.
Figure 4.7: Time-series of the configurational entropy of a stochastic-dynamics replica-exchange simulation of (non-interacting) butane using an exchange frequency of 100 ps$^{-1}$ and eleven replicas spaced evenly between 100 K and 200 K (solid lines). Results from independent stochastic-dynamics simulations at 100 K, 160 K, and 200 K are shown using dashed lines.
Figure 4.8: Time-series of the root-mean-square deviation (rmsd) from the average dihedral angle for the 200 non-interacting butanes from independent stochastic dynamics simulations (dashed lines) at 100K (black), 160K (orange), and at 200K (maroon) and from a replica-exchange stochastic-dynamics simulation using an exchange frequency of 100 ps$^{-1}$ and eleven replicas at temperatures spaced evenly between 100K and 200K (solid lines).
4.8 Bibliography


Chapter 5

Multigraining: an algorithm for simultaneous fine-grained and coarse-grained simulation of molecular systems

5.1 Summary

A method to combine fine-grained and coarse-grained simulations is presented. The coarse-grained particles are described as virtual particles defined by the underlying fine-grained particles. The contribution of the two grain levels to the interaction between particles is specified by a grain-level parameter $\lambda$. Setting $\lambda = 0$ results in a completely fine-grained simulation, whereas $\lambda = 1$ yields a simulation governed by the coarse-grained potential energy surface with small contributions to keep the fine-grained covalently bound particles together. Simulations at different $\lambda$-values may be coupled using the replica-exchange molecular dynamics method (REMD) to achieve enhanced sampling at the fine-grained level.
5.2 Introduction

Most molecular simulations are making use of atom-level (fine-grained) models. This limits the time scale of such simulations for solvated macromolecules to the nanosecond range. Longer time scales can be reached by treating molecules or molecular fragments as single particles or beads, whose motion is simulated using a simple force field describing interbead interactions. When the energy function of such a coarse-grained model is chosen to be smooth and short-ranged, the efficiency of coarse-grained simulations can be orders of magnitude higher than that of corresponding fine-grained simulations, be it at the expense of the loss of atomic detail and some accuracy\textsuperscript{1-5}.

A recently proposed coarse-grained (CG) model\textsuperscript{6} for lipid simulations has the same functional form as the GROMOS force field\textsuperscript{7} except for the use of a switching function\textsuperscript{8} for the non-bonded Lennard-Jones and electrostatic interactions at distances just below the cutoff distance. This model has been implemented into the GROMOS05 simulation package\textsuperscript{9-11}, using a slightly different switching function.

The coarse-grained model has been designed for speed, accuracy, applicability and versatility, where the accuracy is maximized by matching coarse-grained results to fine-grained (FG) simulations as much as possible\textsuperscript{6}. In Figure 5.1 the mapping between the fine-grained (atomistic) and coarse-grained models of hexadecane is shown. Four fine-grained particles may be represented by one coarse-grained particle, located at the centre of mass of the fine-grained particles. One coarse-grained water particle represents four fine-grained water molecules.

The similarity of the coarse-grained and the fine-grained models, in terms of force-field interaction functions and structures suggests, that a combination of the two models into one simulation may be feasible. This would allow for a continuous switching between CG and FG levels of modelling, which would enable both relaxation of large molecular systems and sampling of slow processes with concurrent FG representation of the results.

In the next section a method to combine coarse-grained and fine-grained simulations is presented, followed by results obtained from liquid octane simulations. In the end, merits and shortcomings of the method are discussed.

5.3 Method

5.3.1 Multi-graining Hamiltonian using mapping of coarse-grained particles onto fine-grained ones

Classical molecular dynamics (MD) simulations are represented by the Hamiltonian

\[ H(r,p) = K(p) + U(r), \] (5.1)
Figure 5.1: Hexadecane is shown in a fine-grained and in a coarse-grained representation. Four united atoms (CH₃ or CH₂) of the fine-grained molecule correspond to one coarse-grained alkane particle, and four fine-grained water molecules are represented by just one coarse-grained water particle. The mapping is defined by taking the centre of mass of the fine-grained particles.
where we have used the short-hand notation \( \mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N) \) to indicate a \( N \)-particle configuration with \( \mathbf{r}_i \) the position, \( \mathbf{p}_i \) the momentum and \( m_i \) the mass of particle \( i \), \( K = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} \) the kinetic energy and \( U(\mathbf{r}) \) an interaction energy term representing the interactions between the particles. Generally, one can distinguish between (i) interactions of particles with covalent linkage, i.e. bonds, bond angles and dihedral angles, referred to as bonded interactions, and (ii) interactions between all atom pairs, i.e. the Coulombic interaction and the van der Waals interaction, referred to as non-bonded interactions. In a multi-grained simulation, the interaction energy term consists of four individual terms:

i. the fine-grained bonded interactions \( U_{fg,\text{bonded}}(\mathbf{r}^{fg}) \),

ii. the fine-grained non-bonded interactions \( U_{fg,\text{nonb}}(\mathbf{r}^{fg}) \),

iii. the coarse-grained bonded interactions \( U_{cg,\text{bonded}}(\mathbf{r}^{cg}) \), and

iv. the coarse-grained non-bonded interactions \( U_{cg,\text{nonb}}(\mathbf{r}^{cg}) \), with \( \mathbf{r}^{fg} \) representing the \( N^{fg} \) fine-grained and \( \mathbf{r}^{cg} \) the \( N^{cg} \) coarse-grained particle positions.

The aim of this work is to introduce a parameter \( \lambda \) to this interaction energy term which allows for a continuous change from a fine-grained to a coarse-grained simulation. To this effect, first the \( j = 1, ..., N^{cg} \) coarse-grained particle positions \( \mathbf{r}^{cg}_j \) are defined in terms of fine-grained particle positions:

\[
\mathbf{r}^{cg}_j = g_j(\mathbf{r}^{fg}) = \frac{1}{m_j^{cg}} \sum_{q=1}^{N^{fg}} m_q^{fg} \mathbf{r}_{q}^{fg},
\]

where \( q \) runs over all \( N^{fg} \) fine-grained particles with mass \( m_q^{fg} \) that are mapped (through their centre of mass) to the coarse-grained particle \( j \) (with mass \( m_j^{cg} \)). Thus, the coarse-grained particle positions are not independent dynamical variables. Again, we use below the short-hand notation \( g(\mathbf{r}^{fg}) \equiv (g_1(\mathbf{r}^{fg}), g_2(\mathbf{r}^{fg}), ..., g_{N^{fg}}(\mathbf{r}^{fg})) \).

Combining the single interaction energy terms given above using the grain-level parameter \( \lambda \) and Equation 5.2 into a multi-grained interaction energy term yields

\[
U^{mg}(\mathbf{r}^{fg}; \lambda) = U_{fg,\text{bonded}}(\mathbf{r}^{fg}; \lambda) + (1 - \lambda) U_{fg,\text{nonb}}(\mathbf{r}^{fg}; \lambda) + \lambda U_{cg,\text{bonded}}(g(\mathbf{r}^{fg}); \lambda) + \lambda U_{cg,\text{nonb}}(g(\mathbf{r}^{fg}); \lambda).
\]

Here, the individual interaction energy contributions are defined as functions of \( \lambda \) as they may include \( \lambda \)-dependent force constants or soft-core interactions\(^9,10,12\). Clearly, at \( \lambda = 0 \), standard fine-grained simulations are recovered, whereas setting \( \lambda = 1 \) results in a coarse-grained simulation with the addition of bonded interactions holding the underlying particles of the fine-grained
model together and a slightly different inertia of the coarse-grained particles. From Equation 5.3 the forces on the \( q = 1..N^{fg} \) fine-grained particles are given by

\[
f_q^{fg}(\lambda) = -\frac{\partial}{\partial r_q^{fg}}U^{mg\,fg}(r^{fg};\lambda) = f_q^{fg,\text{bonded}}(r^{fg};\lambda) + (1 - \lambda)f_q^{fg,\text{nonb}}(r^{fg};\lambda) + \lambda \sum_{j=1}^{N^{cs}} \left( f_j^{cs,\text{nonb}}(r^{cs};\lambda) + f_j^{cs,\text{bonded}}(r^{cs};\lambda) \right) \frac{\partial}{\partial r_q^{cs}} g_j(r^{fg}) = f_q^{fg,\text{bonded}}(r^{fg};\lambda) + (1 - \lambda)f_q^{fg,\text{nonb}}(r^{fg};\lambda) + \lambda \sum_{j=1}^{N^{cs}} \left( f_j^{cs,\text{nonb}}(r^{cs};\lambda) + f_j^{cs,\text{bonded}}(r^{cs};\lambda) \right) \frac{m_q^{fg}}{m_j^{cs}} \delta_{qj} \quad (5.4)
\]

where

\[
f_q^{fg,\text{nonb}}(r^{fg};\lambda) = -\frac{\partial}{\partial r_q^{fg}}U^{fg,\text{nonb}}(r^{fg};\lambda),
\]

\[
f_q^{fg,\text{bonded}}(r^{fg};\lambda) = -\frac{\partial}{\partial r_q^{fg}}U^{fg,\text{bonded}}(r^{fg};\lambda),
\]

\[
f_j^{cs,\text{nonb}}(r^{cs};\lambda) = -\frac{\partial}{\partial r_j^{cs}}U^{cs,\text{nonb}}(r^{cs};\lambda),
\]

\[
f_j^{cs,\text{bonded}}(r^{cs};\lambda) = -\frac{\partial}{\partial r_j^{cs}}U^{cs,\text{bonded}}(r^{cs};\lambda),
\]

and \( \delta_{qj} \) is 1 if the fine-grained particle \( q \) is one of the defining particles (through the mapping of Equation 5.2) of the coarse-grained particle \( j \) and 0 otherwise.

In Figure 5.2 the algorithm of a single (leap-frog) integration step during a multi-grained simulation is depicted. The left hand side represents the fine-grained level (using a fine-grained topology and fine-grained particles with positions \( r_q^{fg} \) and momenta \( p_q^{fg} \)), the right hand side corresponds to the coarse-grained level (with coarse-grained topology and configuration \( r_j^{cs} \)). During the integration step, first the coarse-grained positions are updated from the fine-grained positions using the virtual-grain definition (usually defined as one coarse-grained particle being the centre of mass of a given number, e.g. four, of fine-grained particles, see Equation 5.2). Then, energies and forces are calculated at both levels, using the respective parts of the interaction energy terms \( (U^{fg}(r^{fg};\lambda) \text{ and } U^{cs}(r^{cs};\lambda)) \). Afterwards, the potential energy of the multi-grained simulation is calculated from the individual terms and the fine-grained forces are augmented by the coarse-grained ones according to Equation 5.4 (last term), weighted by the grain-level parameter \( \lambda \). Finally, the fine-grained particle positions and momenta are propagated forward in time using the leap-frog integration scheme.
Figure 5.2: The algorithmic sequence of operations during one (leap-frog) integration step of a multi-grained simulation is shown. The left half corresponds to operations applied to the fine-grained part of the simulation, the right half to operations acting on the coarse-grained part. Twice during the integration step, one switches between the two grain levels. First, the coarse-grained particle positions are computed from the fine-grained ones (step 1). Then, forces and energies calculated in the coarse-grained representation are distributed back onto the underlying fine-grained particles (step 3). Step 4b is only carried out for non-mapped coarse-grained particles.
5.3. Method

Using this procedure, it is possible to smoothly change from a fine-grained simulation into a coarse-grained one and back. Therefore, a technique to achieve fast equilibration of a fine-grained system would consist of increasing \( \lambda \) to 1, equilibrating at the coarse-grained level, and then, in the end reducing \( \lambda \) to 0 again. An example of this technique is given in the second part of Section 5.4.

5.3.2 Replica-exchange multigraining simulation

The full strength of the multigraining method lies in its combination with replica exchange MD (REMD\textsuperscript{14-23}, also known as parallel tempering). This method has been generalized\textsuperscript{24-26} to simulate a set of \( N_s \) replicas, each governed by a different Hamiltonian \( H(r^{fg,s}, p^{fg,s}; \lambda^s) \) with \( s = 1..N_s \), and with \( r^{fg,s} \) the (fine-grained) positions, \( p^{fg,s} \) the (fine-grained) momenta, and \( \lambda^s \) the \( \lambda \) of replica \( s \). The probability of the global state \( S' \) consisting of the \( N_s \) replicas is proportional to its weight factor

\[
W(S') = \exp \left( - \sum_{s=1}^{N_s} \beta H(r^{fg,s}, p^{fg,s}; \lambda^s) \right),
\]

with \( \beta = 1/k_BT, k_B \) is Boltzmann's constant and \( T \) the temperature. To sample canonical ensembles for all the different Hamiltonians \( H(r^{fg,s}, p^{fg,s}; \lambda^s) \) even if from time to time two replicas are exchanged, the criterion of detailed balance has to be fulfilled for the replica exchanges. Detailed balance, with \( w(S' \rightarrow S'') \) the transition probability from (global) state \( S' \) to state \( S'' \), and \( s', s'' \) the indices of the two replicas in \( S' \) that are exchanged to obtain \( S'' \) from \( S' \), is given as

\[
W(S')w(S' \rightarrow S'') = W(S'')w(S'' \rightarrow S').
\]

This condition can be satisfied through the usual Metropolis criterion\textsuperscript{27} for the probability \( p(s' \leftrightarrow s'') \) of an exchange of the two replicas \( s' \) and \( s'' \),

\[
p(s' \leftrightarrow s'') = \frac{w(S'' \rightarrow S')}{W(S')}, \quad w(S' \rightarrow S'') = \begin{cases} 1 & \text{for } \Delta \leq 0, \\ \exp(-\Delta) & \text{otherwise,} \end{cases}
\]

with

\[
\Delta = \beta \left( U_{mg}(r^{fg,s''}; \lambda^{s''}) - U_{mg}(r^{fg,s'}; \lambda^{s'}) \right) - \beta \left( U_{mg}(r^{fg,s'}; \lambda^{s'}) - U_{mg}(r^{fg,s''}; \lambda^{s''}) \right).
\]

Note that before every exchange attempt, the potential energy \( U^{mg}(r^{fg}; \lambda) \) has to be evaluated for both replicas at both \( \lambda^{s'} \) and \( \lambda^{s''} \) values of the coupling parameter \( \lambda \).

While it is not possible to simulate very large systems using this method, as all the fine-grained degrees of freedom are, irrespective of the value of \( \lambda \), explicitly treated, the fine-grained
simulation of the replicas at \( \lambda = 0 \) can benefit from the faster sampling obtained in the coarser-grained simulation of the replicas at larger \( \lambda \). Coarse graining a simulation enhances the sampling in two ways. First, by smoothening out the potential energy landscape transitions from one local minimum into another one get more likely. Second, this smoother potential energy surface allows the time-step to be increased by a factor of 15 to 20 for typical coarse-grained models of condensed phase molecular systems. As the fine-grained bonded interactions are retained even at high \( \lambda \) values (more weight for coarse-grained forces), just increasing the time-step in a multigraining simulation is not possible. Therefore, multiple time-stepping is used to separate the often calculated, but computationally cheap fine-grained bonded forces from the expensive nonbonded ones\(^{28}\).

### 5.3.3 Multi-graining Hamiltonian with partial mapping of coarse-grained particles onto fine-grained ones

The coarse-grained model used here\(^6\) maps (usually) four fine-grained particles (united atoms or molecules) on one coarse-grained particle. That way, simple alkanes and even more complex lipids can be represented at coarse grain level and defining the coarse-grained particles as virtual particles using fine-grained positions is straightforward. This approach does not work for solvent molecules such as water, as one coarse-grained solvent represents many (usually four) fine-grained solvent molecules, which may diffuse away from each other. There are (at least) three ways to overcome this problem.

First, the most simple technique would be to hold four fine-grained solvent molecules together by using (four to six) loose distance restraints between them. Then, the coarse-grained solvent particle could still be mapped on the centre of mass of the restrained solvent molecules. This approach has not been investigated further yet, as minimal impact on the fine-grained simulation was a priority throughout this work.

In a second approach, fine-grained and coarse-grained solvent particles could be treated independently corresponding to a configuration defined by \((r^{\text{solv}^{\text{fg}}}, r^{\text{solv}^{\text{fg}}}, r^{\text{solv}^{\text{fg}}}, r^{\text{solv}^{\text{fg}}})\), where \(r^{\text{solv}^{\text{fg}}}\) is mapped onto \(r^{\text{solv}^{\text{fg}}}\), but \(r^{\text{solv}^{\text{fg}}}\) is not mapped onto \(r^{\text{solv}^{\text{fg}}}\). In other words, the non-mapped solvent particles are treated as additional degrees of freedom in the Hamiltonian. For these non-mapped solvent particles the coarse-grained interaction function of Equation 5.3 depending on the fine-grained solvent degrees of freedom (with configuration \(r^{\text{solv}^{\text{fg}}}\)) is set to zero,

\[
U^{\text{cg,bonded,solv}^{\text{fg}}} (g(r^{\text{fg}}), g(r^{\text{solv}^{\text{fg}}}); \lambda) = 0,
\]

\[
U^{\text{cg,nonb,solv}^{\text{fg}}} (g(r^{\text{fg}}), g(r^{\text{solv}^{\text{fg}}}); \lambda) = 0,
\]  

(5.10)

and the degrees of freedom for the non-mapped coarse-grained solvent particles (with configura-
5.4 Results

A system of 128 octane molecules was simulated using periodic boundary conditions. The starting configuration was separated into two layers, the first 64 octanes belonging to the lower, the
Fine-grained simulations were carried out for 500 ps using GROMOS05 with the GROMOS 45A3 force-field parameter set, an integration time-step of 2 fs, at constant volume and a temperature of 300 K (weakly coupled; $\tau = 0.1$ ps). Centre of mass translation was strictly removed, covalent bond energies and forces were calculated using a harmonic potential energy function (GROMOS87), and a triple range cutoff scheme was applied with a short range cutoff of 0.8 nm, a long range cutoff of 1.4 nm and a reaction-field dielectric permittivity $\varepsilon$ of 1.0.

Coarse-grained simulations were carried out for approximately 500 ps using GROMOS05 with the coarse-grained model developed recently by Marrink et al., an integration time-step size of 30 fs, at constant volume and at a temperature maintained at 300 K by weak coupling ($\tau = 0.1$ ps) to a temperature bath. Centre of mass translation was strictly removed, bond energies and forces were calculated using, in accordance to the coarse-grained model, $\varepsilon = 20$, a harmonic potential energy function (GROMOS87), and the Lennard-Jones potential energy was smoothly shifted to zero at a cutoff of 1.2 nm.

In Figure 5.3 the average over all molecules of the $z$-component of the position of every fourth octane atom (fine-grained) or of every first octane atom (coarse-grained) belonging initially to the upper or lower layer are shown. The potential energy hypersurfaces governing the diffusion controlled mixing of the two octane layers seem to be quite equivalent for the fine-grained and the coarse-grained models, considering their similar mixing behaviour. Of course, the fine-grained one is rougher on a small time scale, so that a much shorter integration time-step of 2 fs had to be used, compared to the 30 fs time-step in the coarse-grained simulation. Therefore, the latter one relaxes computationally at least 15 times more efficiently.

The same system was also simulated multi-grained using the grain level parameter $\lambda$ in a replica-exchange MD framework. 24 replicas were used with the following $\lambda$ values: 0.00, 0.08, 0.14, 0.20, 0.25, 0.30, 0.34, 0.38, 0.42, 0.46, 0.50, 0.54, 0.58, 0.62, 0.66, 0.70, 0.74, 0.78, 0.82, 0.86, 0.90, 0.93, 0.96 and 1.00. In this setup, the average over time and pairs of replicas of the switching probability was 26%, ranging from 8% up to 42% for the different pairs of $\lambda$-values. All replicas were started from an identical structure and for the first 20 exchange trials, switching between replicas was prohibited. In between two exchange trials, the replicas were independently evolved for 500 steps, using a time-step size of 2 fs for $\lambda < 0.55$, 10 fs for $0.55 \leq \lambda < 0.9$, 20 fs for $0.90 \leq \lambda < 1.00$ and 30 fs for $\lambda = 1.00$. In other words, at least 1 ps of independent evolution separated the exchange trials. Figure 5.4 shows the grain-level ($\lambda$) time-series of each replica. The 10 replicas with initially the lowest $\lambda$-values are marked with bold lines, a couple of interesting ones in colour. As the simulated system is relatively simple and all replicas are starting from the same configuration, it happens twice that a number of replicas are very close in potential energy. In this case, the Metropolis criterion (Equation 5.8) yields nearly 100% exchange probability, therefore these two intervals are marked by numerous replica exchanges. Additionally, due to the artificial setup of the computational box with separation of the octane molecules into
Figure 5.3: The average over all molecules of the z-component of the fourth atom (fine-grained) or the first atom (coarse-grained) of the octane molecules belonging to the (initially) upper layer and the ones belonging to the (initially) lower layer versus time during an MD simulation of 500 ps length are shown. Black lines correspond to the fine-grained, blue lines to the coarse-grained simulation.

two layers, the system is not at equilibrium at the start of the simulations. Therefore, it is to be expected that the total potential energy will decrease slightly during the simulation. As this happens faster for the more coarse-grained replicas, they are likely to switch with more fine-grained replicas, explaining why most of the initially fine-grained replicas end up at high grain-level at the end of this short REMD simulation.

Figure 5.5 shows the average over all molecules of the z-component of the position of every fourth octane atom (using the atomistic representation) during the REMD simulation. The black line and the dashed black line correspond to molecules of the replicas at grain-level $\lambda = 0.0$ initially in the upper and lower layers respectively, the blue lines to molecules of the replicas at $\lambda = 1.0$ initially in the upper and lower layers respectively. The lower panel indicates successful replica exchanges involving the replicas at $\lambda = 0.0$, where the starting grain-level of the replica with which the switch is made, is indicated in some cases. Using this method, equilibration of
Figure 5.4: Time evolution of the grain-level ($\lambda$) of the 24 replicas during 300 replica exchange steps of a multi-grained REMD simulation of octane. The 10 replicas with initially the lowest $\lambda$-values are marked by bold lines, selected interesting replicas by coloured lines.

the two layers, at fine-grained level, can be achieved within 100 ps, even though in the first 20 ps, no exchanges are allowed. At coarse-grained level this fine-grained time period corresponds already to 1.5 ns of simulation.

As a second example, 25 hexadecane molecules were simulated in SPC water$^{32}$. The system was simulated multi-grained at grain-level $\lambda = 1.0$. Fine-grained and coarse-grained solvent particles were not mapped, but 875 of the latter were added to the multi-grained Hamiltonian as described in Equations 5.11 and 5.12. Rectangular periodic boundary conditions were applied. The temperature of the solutes and the solvent were separately maintained at 300 K by weakly coupling to a temperature bath, while the volume of the box was fixed. The GROMOS 45A3$^{29}$ parameter set was used for the fine-grained model, the coarse-grained model corresponded to that of Marrink et al.$^{6}$. Multiple time-stepping$^{28}$ was used with a short time-step of 2 fs for the fine-grained bonded interactions and a large time-step of 30 fs for the coarse-grained interaction terms. Two configurations, frames 40 (8 ps) and frame 125 (25 ps) were selected to serve as starting points for independent short multi-grained simulations, continuously changing the grain-
5.4. Results

Figure 5.5: The average over all molecules of the z-component of the fourth atom (using the fine-grained representation of the multi-grained system) of the octane molecules belonging (initially) to the upper layer and the ones belonging (initially) to the lower layer versus replica exchange trials during a REMD simulation covering 300 exchange trials (500 steps per trial) are shown. Black lines correspond to the replicas at grain-level $\lambda = 0.0$ (FG), blue lines to the replicas at $\lambda = 1.0$ (CG). In the lower panel, successful exchanges of the replicas at $\lambda = 0.0$ are indicated and for some, the original (starting) grain-level of the exchange-partner replica is indicated.

level from 1.0 to 0.0 within 0.5 ps, with an integration time-step of 2 fs. In Figure 5.6 snapshots of the system are depicted. From left to right, the starting structure in its multi-grained representation, then frame 40 and 125 in the coarse-grained representation, and finally the end-state (100 ps), with the aggregated hexadecanes again in multi-grained representation are shown. In the middle, the final conformations after the short multi-grained simulations in which the grain-level was changed from 1.0 to 0.0 are shown. These conformations may now be used for further fine-grained simulation and analysis.
Figure 5.6: Multi-grained simulation of 25 hexadecanes in explicit SPC\textsuperscript{32} water. The first snapshot shows the multi-grained representation of the starting structure, then, in the top row, frame 40 (8 ps) and 125 (25 ps) in coarse-grained representation and, to the right, the final, aggregated configuration is shown. From frame 40 and frame 125, short (0.5 ps) multi-grained simulations changing the grain-level from 1.0 (CG) to 0.0 (FG) were done and their final configurations are shown in fine-grained representation.

5.5 Discussion

A method to couple simulations at fine-grained and at coarse-grained level was presented. At 100\% fine-grained level (grain-level parameter $\lambda = 0.0$), a standard fine-grained simulation is retained, whereas at the coarse-grained level ($\lambda = 1.0$), in addition to the potential energy terms from the coarse-grained model, the fine-grained particles, which are covalently bonded together, must be kept together. The advantage of a large integration time-step at coarse-grained level may, nevertheless, be maintained by using a multiple time-step approach (such as reversible RESPA\textsuperscript{28}).

The method may be used to achieve fast equilibration, in a fashion similar to simulated annealing, by increasing the grain-level ($\lambda$), simulating the now coarse-grained system and then decreasing the grain-level again to the fine-grained level.

A second application is in replica-exchange simulations, where the grain-level parameter $\lambda$ is used to distinguish between the different replicas. It was shown that fast equilibration at the fine-grained level can be reached, and additionally, predictions of the long term behaviour of the system may be extracted from the simulations at high grain-level.

Note that using this method, there is no need to reconstruct fine-grained particle positions.
from coarse-grained ones and all movements are governed by a (time-independent) Hamiltonian. The disadvantage is, that the system size limit due to fine-grained modelling is not circumvented with this method. But, sampling of the configurational space is greatly enhanced by replica exchange using replicas governed by a smoother potential energy surface (coarse-grained) and therefore simulated with a much larger time-step size.

Suggestions of how to overcome the difficulties induced by the diffusive nature of solvent molecules in a multi-grained simulation have been made and are currently investigated.

5.6 Acknowledgements

We would like to thank Andrew E. Torda for an interesting discussion at the beginning of this work. Financial support by the National Center of Competence in Research (NCCR) Structural Biology of the Swiss National Science Foundation (SNSF) is gratefully acknowledged.
5.7 Bibliography


5.7. Bibliography


Chapter 6

Sampling of rare events using hidden restraints

6.1 Summary

A method to enhance sampling of rare events is presented. It makes use of distance or dihedral-angle restraints to overcome an energy barrier separating two metastable states or to stabilize a transition state between the two metastable states. In order not to perturb these metastable end states themselves, a prefactor is introduced into the restraining energy function, which smoothly increases the weight of this function from zero to one at the transition state or on top of the separating energy barrier and then decreases the weight again to zero at the final state. The method is combined with multi-configurational thermodynamic integration and applied to two biomolecular systems which were difficult to treat using standard thermodynamic integration. As first example the free energy difference of a cyclic α-aminoxy-hexapeptide-ion complex upon changing the ion from Cl\(^-\) to Na\(^+\) was calculated. A large conformational rearrangement of the peptide was necessary to accommodate this change. Stabilizing the transition state by (hidden) restraints facilitates that. As second example the free energy difference between the \(^4C_1\) and the \(^1C_4\) conformation of β-D-glucopyranoside was calculated. In unrestrained simulations the change from the \(^4C_1\) into the \(^1C_4\) conformation was never observed, because of the high energy barrier separating the two states. Using (hidden) restraints, the transition from the \(^4C_1\) into the \(^1C_4\) state and back could be enforced without perturbing the end states. As comparison, for the same transitions the potential of mean force as obtained by using dihedral-angle constraints is provided.
6.2 Introduction

Computer simulation is increasingly used to investigate in atomic detail dynamic molecular processes. Atomistic molecular dynamics (MD) simulations of systems comprising tens to even hundredths of thousands of atoms are nowadays possible covering nanoseconds to microseconds. Yet, conformational changes in biomolecules, ligand binding or structural organization occur roughly in the millisecond range, while protein folding may take up to minutes. This means that these interesting events in biomolecular systems are often not or only rarely observed in MD simulations. Therefore, methods to enhance the sampling of these rare events during an MD simulation have been developed over the years\textsuperscript{1-3}. Often, one is only interested in a few selected degrees of freedom of the system under the average influence of the many other degrees of freedom. Then, statistical mechanics provides means of expressing the average dependence of the degree of freedom of interest on the residual degrees of freedom in the form of a potential of mean force (PMF)\textsuperscript{4}. The potential of mean force can be defined as minus $k_B T$ times the logarithm of the probability that the system is found at a specific position along the degree of freedom under investigation (at temperature $T$ and with $k_B$ being the Boltzmann constant), and it can be interpreted as a projection of the free energy on one (or more) coordinates or degrees of freedom of interest.

Using MD simulation, the potential of mean force may be calculated by constraining the system to a specified point along the investigated degree of freedom and calculating the average value of the constraint force magnitude\textsuperscript{5-7}. Another way to calculate a potential of mean force is by using umbrella sampling\textsuperscript{8}. Sampling is biased towards otherwise unfavorable regions of the configurational space by performing simulations with an additional artificial biasing potential energy term, the so-called umbrella potential. The resulting probability distribution from the simulation of the unphysical system can be corrected afterwards to yield the probability distribution of the corresponding physical, unbiased, system. The biasing potential may be used to focus sampling in a specific region, a so-called window, along the degree of freedom for which the potential of mean force is to be determined. By performing multiple simulations with shifted focuses, the complete degree of freedom can be sampled segment by segment. Each segment of the potential of mean force calculated from a different window, has an arbitrary offset. To obtain the resulting, continuous potential of mean force, the segments must be matched. A third method to determine a potential of mean force is a combination of the two mentioned above: the biasing force is measured from which the PMF can be constructed\textsuperscript{9}.

All three ways to calculate the potential of mean force, which may then be integrated to get the free energy difference between two states, the so-called end states, on the degree of freedom considered, suffer from the disadvantage, that the biasing or restraining (or constraining) potential energy term is also applied in the two end states. This may lead to artifacts in the relative free energies, as no longer the difference between two unrestrained metastable states, but only the difference between two restrained states is calculated. In other words, the free energy difference
6.3 Method

If a molecular system exhibits two (meta) stable states which are connected by a transition path that is only very rarely sampled during an MD simulation\(^{10}\), it may be efficient to forcefully propagate the system from one end state to the other along this transition path. To be able to calculate an unperturbed relative free energy of the one state with respect to the other it is of advantage not to put any external forces or restraints on the system when being in these two end states. Therefore, the following general restraint formulation is proposed to enforce a transition from state A to B without influencing the end states:

\[
q^{\text{restr}}(r; \lambda) = 2^{n+m}\lambda^n (1 - \lambda)^m q^{AB}_{\text{restr}}(r; \lambda), \tag{6.1}
\]

where \(q^{AB}_{\text{restr}}(r; \lambda)\) is a \(\lambda\) dependent (restraining) potential energy term which enforces the (smooth) transition of the system from state A into state B along the pathway \(\lambda = 0\) to \(\lambda = 1\). The weight factor \((2^{n+m}\lambda^n (1 - \lambda)^m)\) applied to the restraining potential energy term \(V_{\text{restr}}^{AB}\) is shown in Figure 6.1.

The relative free energy \(\Delta G_{BA}\) of state B with respect to state A may be calculated using the (multi-configurational) thermodynamic integration method\(^{11}\) (reviewed elsewhere\(^{12,13}\))

\[
\Delta G_{BA} = \int_{\lambda=A}^{\lambda=B} \left< \frac{\partial H}{\partial \lambda} \right>_{\lambda} d\lambda, \tag{6.2}
\]

with \(H\) being the Hamiltonian of the system, which also includes the restraining potential energy term \(q^{\text{restr}}\). The ensemble (or time) average of the derivative of the Hamiltonian with respect to \(\lambda\), \(\left< \frac{\partial H}{\partial \lambda} \right>_{\lambda}\), is calculated for a given number \(N_\lambda\) of fixed \(\lambda\) values using \(N_\lambda\) MD simulations, while the integration is carried out numerically afterwards.
Figure 6.1: Weight factor $2^{n+m} \lambda^n (1 - \lambda)^m$ as function of $\lambda$ for different values of the exponents $n$ and $m = n$. The solid line corresponds to $n = 1$, the dashed line to $n = 2$, the dashed-dotted line to $n = 3$ and the dotted line to $n = 4$.

The $\lambda$ derivative of the restraining term $V_{\text{restr}}$ is

$$\frac{\partial}{\partial \lambda} V_{\text{restr}}(\mathbf{r}) = 2^{n+m} \left[ (n\lambda^{n-1}(1-\lambda)^m - m\lambda^n(1-\lambda)^{m-1}) V_{\text{restr}}(\mathbf{r}; \lambda) + \lambda^n(1-\lambda)^m \frac{\partial}{\partial \lambda} V_{\text{restr}}^{AB}(\mathbf{r}; \lambda) \right],$$

and the force is

$$f_i = -\frac{\partial}{\partial \mathbf{r}_i} V_{\text{restr}}(\mathbf{r}; \lambda) = -2^{n+m} \lambda^n (1-\lambda)^m \frac{\partial}{\partial \mathbf{r}_i} V_{\text{restr}}^{AB}(\mathbf{r}; \lambda),$$

where $\mathbf{r}$ denotes a configuration of the system and $\mathbf{r}_i$ the coordinates of particle $i$. The method may be used with any kind of restraining potential energy function $V_{\text{restr}}(\mathbf{r}; \lambda)$. The following two sections show distance and dihedral-angle restraints as examples.
6.3. Method

6.3.1 Distance restraints

A distance restraint \( k \) between atoms \( k_1 \) and \( k_2 \) may be formulated using a harmonic potential energy function. In practice, it is often necessary to smoothly linearize the potential energy function beyond a specified distance \( r_{in}^0 \) to avoid very high energies and extremely steep slopes,

\[
q_{disres}^k(r) = \begin{cases} 
-K_k^{\text{dist}} \left( |r_k| - r_k^0 + 1/2r_{in}^0 \right) r_{in}^0 & |r_k| < r_k^0 - r_{in}^0 \\
1/2K_k^{\text{dist}} \left( |r_k| - r_k^0 \right)^2 & r_k^0 - r_{in}^0 \leq |r_k| \leq r_k^0 + r_{in}^0 \\
K_k^{\text{dist}} \left( |r_k| - r_k^0 - 1/2r_{in}^0 \right) r_{in}^0 & |r_k| > r_k^0 + r_{in}^0 
\end{cases} \tag{6.5}
\]

where \( K_k^{\text{dist}} \) is the force constant of the restraint, \( r_k = r_k - r_k \) is the vector connecting particle \( k_2 \) to particle \( k_1 \) and the restraint is linearized for distances larger than \( r_{in}^0 \).

If the distance restraint should drive the system (in a controlled way) from a state \( A \) to a state \( B \), the restraint parameters have to change along this path. This change of restraint parameters along the path described by \( \lambda \) can be formulated as

\[
q_{disres,AB}^{\text{harm},k}(r; \lambda) = 1/2 \left( (1 - \lambda)K_k^A + \lambda K_k^B \right) \left( |r_k| - (1 - \lambda)r_k^0 - \lambda r_k^0 \right)^2 \tag{6.6}
\]

for the harmonic part and as

\[
q_{disres,AB}^{\text{lin},k}(r; \lambda) = \zeta \left( (1 - \lambda)K_k^A + \lambda K_k^B \right) \left( |r_k| - (1 - \lambda)r_k^0 - \lambda r_k^0 \right) r_{in}^0 \tag{6.7}
\]

for the linearized part (with \( \zeta = -1 \) if \( |r_k| < r_k^0 - r_{in}^0 \) and \( \zeta = 1 \) if \( |r_k| > r_k^0 + r_{in}^0 \)) using a restraint length in state \( A \) of \( r_k^A \) and in state \( B \) of \( r_k^B \) and corresponding force constants \( K_k^A \) and \( K_k^B \). The force in the harmonic part is

\[
f_i = -\frac{\partial}{\partial r_i} q_{disres,AB}^{\text{harm},k}(r; \lambda) = -\left( (1 - \lambda)K_k^A + \lambda K_k^B \right) \left( |r_k| - (1 - \lambda)r_k^0 - \lambda r_k^0 \right) \frac{r_k}{|r_k|} (\delta_{ik_1} - \delta_{ik_2}) \tag{6.8}
\]

and in the linearized part

\[
f_i = -\frac{\partial}{\partial r_i} q_{disres,AB}^{\text{lin},k}(r; \lambda) = -\zeta \left( (1 - \lambda)K_k^A + \lambda K_k^B \right) r_{in}^0 \frac{r_k}{|r_k|} (\delta_{ik_1} - \delta_{ik_2}), \tag{6.9}
\]

where \( \delta \) is the Kronecker delta.

Finally, to use the thermodynamic integration formula (6.2), the derivative of the distance restraint with respect to integration variable \( \lambda \) needs to be known:

\[
\frac{\partial}{\partial \lambda} q_{disres,AB}^{\text{harm},k}(r; \lambda) = 1/2 \left( \left( K_k^B - K_k^A \right) \left( |r_k| - (1 - \lambda)r_k^0 - \lambda r_k^0 \right)^2 + \\
2 \left( (1 - \lambda)K_k^A + \lambda K_k^B \right) \left( |r_k| - (1 - \lambda)r_k^0 - \lambda r_k^0 \right) \left( r_k^0 - r_k^B \right) \right), \tag{6.10}
\]

where \( \lambda \) is the integration variable.
Chapter 6. Sampling of rare events using hidden restraints

and

\[
\frac{\partial}{\partial \lambda} V_{lin,k}^{disres,AB}(\mathbf{r}; \lambda) = \zeta \left( (K_k^B - K_k^A) \left( |r_k| - (1 - \lambda) r_{k0}^A - \lambda r_{k0}^B - \frac{1}{2} \frac{\partial V_{lin}^0}{\partial r_{lin}^0} \right) r_{lin}^0 + \right) \left( (1 - \lambda) K_k^A + \lambda K_k^B \right) r_{lin}^0 \left( r_{k0}^A - r_{k0}^B \right).
\]

(6.11)

An example of using distance restraints in a relative free energy calculation is given in Section 6.4.

6.3.2 Dihedral-angle restraints

Many structural differences of biomolecules are related to different dihedral angles, an example being the \( \psi \) and \( \phi \) angles in peptides\(^\text{14} \). Similar to distances, dihedral angles may be restrained using a harmonic potential (for a dihedral angle specified by the atoms \( k_1 - k_2 - k_3 - k_4 \))

\[
\eta^{dihres}(\phi) = \begin{cases} 
-K_k^{dih}(\Delta \phi_k + 1/2 \phi_{lin0}^0) \phi_{lin0}^0 & \Delta \phi_k < -\phi_{lin0}^0 \\
1/2 K_k^{dih}(\Delta \phi_k)^2 & -\phi_{lin0}^0 \leq \Delta \phi_k \leq \phi_{lin0}^0 \\
K_k^{dih}(\Delta \phi_k - 1/2 \phi_{lin0}^0) \phi_{lin0}^0 & \Delta \phi_k > \phi_{lin0}^0 
\end{cases}
\]

(6.12)

with \( K_k^{dih} \) the restraining force constant, \( \phi_{lin0}^0 \) the angle to restrain to and \( \Delta \phi_k = \phi_k - \phi_{lin0}^0 + 2n\pi \), where \( n \) is chosen such that \( \phi \) is within the range \([\phi_{k0}^{max}, \phi_{k0}^{max} + 2\pi]\) and assuming that \( \phi_k^{0} \) is chosen within the same range. Using this dihedral-angle restraint formulation, \( \phi_k^{max} \) determines at which position the direction of the rotation around the dihedral angle, caused by the restraint potential energy function, inverses. By aligning \( \phi_k^{max} \) to the highest potential energy barrier for the rotation around the dihedral angle, it is possible to avoid pushing against this barrier, by rotating the other way instead.

Similar to distance restraining a \( \lambda \) dependence can be introduced to enforce conformational sampling along a pathway from state A to state B

\[
\eta^{disres,AB}_{harm,k}(\phi; \lambda) = 1/2 \left( (1 - \lambda) K_k^A + \lambda K_k^B \right) (\Delta \phi_{k\lambda})^2,
\]

(6.13)

using

\[
\Delta \phi_{k\lambda} = \phi_k - (1 - \lambda) \phi_{k0}^A - \lambda \phi_{k0}^B + 2n\pi,
\]

(6.14)

and

\[
\eta^{disres,AB}_{lin,k}(\phi; \lambda) = \left( (1 - \lambda) K_k^A + \lambda K_k^B \right) (\zeta \Delta \phi_{k\lambda} - 1/2 \phi_{lin0}^0) \phi_{lin0}^0
\]

(6.15)

with \( \zeta = -1 \) if \( \Delta \phi_{k\lambda} < -\phi_{lin0}^0 \) and \( \zeta = 1 \) if \( \Delta \phi_{k\lambda} > \phi_{lin0}^0 \) for the linearized part of the restraint. The parameter \( \phi_k^{max} \) is kept constant along the path, assuming that common maximum values of the
restraining potential energy terms in states A and B can be found. The force in the harmonic part of the restraining potential energy function is

\[ f_i = -\frac{\partial}{\partial \Delta \phi_{kl}} \psi_{\text{dihres},AB} \frac{\partial \phi_k}{\partial \phi_d} \frac{\partial \phi_k}{\partial r_i} \]

\[ = -(1 - \lambda) K_k^A + \lambda K_k^B \Delta \phi_{kl} \frac{\partial \phi_k}{\partial r_i} \]  

(6.16)

and in the linear part

\[ f_i = -\frac{\partial}{\partial \Delta \phi_{kl}} \psi_{\text{dihres},AB} \frac{\partial \phi_k}{\partial \phi_d} \frac{\partial \phi_k}{\partial r_i} \]

\[ = -(1 - \lambda) K_k^A + \lambda K_k^B \zeta \phi_{\text{lin},i} \frac{\partial \phi_k}{\partial r_i} \]  

(6.17)

with \( \frac{\partial \phi_k}{\partial r_i} \) equivalent to the expression used for the (physical) dihedral-angle potential energy term15-17.

Finally, the \( \lambda \) derivative of the restraint is given by

\[ \frac{\partial}{\partial \lambda} \psi_{\text{dihres},AB} \text{harm,} k = \frac{1}{2} \left( (K_k^B - K_k^A) (\Delta \phi_{kl})^2 + 2 \left( (1 - \lambda) K_k^A + \lambda K_k^B \right) \Delta \phi_{kl} \left( \phi_{k}^{A} - \phi_{k}^{B} \right) \right) \]  

(6.18)

and

\[ \frac{\partial}{\partial \lambda} \psi_{\text{dihres},AB} \text{lin,} k = \phi_{\text{lin},i} \left( (K_k^B - K_k^A) \left( \zeta \Delta \phi_{kl} - 1/2 \phi_{\text{lin},i} \right) + \left( (1 - \lambda) K_k^A + \lambda K_k^B \right) \zeta \left( \phi_{k}^{A} - \phi_{k}^{B} \right) \right) \]  

(6.19)

6.4 Applications

In this section, results from studies on two types of problems are shown. First, a cyclic aminoxy-hexapeptide showing the interesting possibility to bind anions and (less tightly) also cations was investigated. When changing from binding an anion to binding a cation, the structure of the peptide has to change dramatically. Standard simulation techniques fail to reproduce this conformational change on a time-scale accessible to MD simulations. Second, the relative stability of chair conformations \( ^1C_4 \text{ and } ^4C_1 \) of a hexopyranose was investigated. During a standard MD simulation, not enough transitions between the two conformations occur to accurately calculate their free energy difference.

The two systems have in common that standard simulation does not appropriately sample the part of the phase space important to calculate the free energy differences between different
metastable conformations of the systems. But, while in the former system the complex between ion and peptide becomes instable upon (alchemically) changing from a cation to an anion and therefore needs stabilization of the transition state, the latter system needs to be forced over the barrier separating conformation A from B, i.e. needs destabilization of the transition state. Both requirements may be fulfilled by adequately choosing (hidden) restraints.

6.4.1 Conformations of a cyclic aminoxy-hexapeptide upon binding cations and anions

Recently, a cyclic D,L-α-aminoxy-hexapeptide with the sequence (D)-Leu - (L)-Phe - (D)-Leu - (L)-Phe - (D)-Leu - (L)-Phe was synthesized and its properties investigated as part of a study of the functions of foldamers, that is, of oligomers of unnatural peptide moieties that fold into well-defined secondary structures. Cyclic D,L-α amino acid peptides and β3-amino acid peptides have been found to self-assemble into nanotubes and function as transmembrane ion channels, while α-aminoxy acids, as backbone analogue of β-amino acids, can form an eight-membered intramolecular hydrogen bonded ring, and, as oligomers 1.8 helices. The cyclic peptide with its small pore size was expected to bind ions. The carbonyl groups may coordinate with cations, whereas the amide hydrogen atoms may form hydrogen bonds with anions (see Figure 6.2). Experiments found only halide ions, but not alkali metal ions, to bind to the cyclic hexapeptide.

Here, conformational changes of the hexapeptide upon changing the complexated ion from \( \text{Cl}^- \) to \( \text{Na}^+ \) were investigated, using the multi-configurational thermodynamic integration method. The complex was simulated by integrating Newton's equation of motion based on the leapfrog scheme in explicit chloroform solvent, at a temperature of 300 K (maintained by separate weak coupling of solute and solvent to a temperature bath with \( \tau_T = 0.1 \text{ ps} \)), and constant pressure of 1 atm (\( \tau_P = 0.5 \text{ ps} \), \( \kappa_P = 4.575 \times 10^{-4} (\text{kJ} \text{mol}^{-1} \text{nm}^{-3})^{-1} \), using isotropic coordinate scaling). Bond lengths were constrained using the SHAKE algorithm with a relative geometric tolerance of \( 10^{-4} \). A triple range cutoff-scheme was used with a short cutoff of 0.8 nm and a long cutoff of 1.4 nm, and a reaction-field approximation (\( \varepsilon_{rf} = 4.81 \)) was applied. Center of mass translation was removed. For interaction parameters the GROMOS 45A3 parameter set of the GROMOS force field was used together with some special aminoxy parameters. In Figure 6.2 configurations of the hexapeptide complexing a chloride and a sodium ion are shown.

The ensemble average of the partial derivative of the Hamiltonian with respect to the (alchemical) \( \lambda \)-dependent pathway from \( \text{Cl}^- \) to \( \text{Na}^+ \) and backwards is depicted in Figure 6.3 and the (numerically) integrated free energy differences \( \Delta G \) are given in Table 6.1. In Figure 6.4, showing the averages of the distances between the backbone carbonyl oxygen and amid hydrogen atoms and the ion during the thermodynamic integration, the problems arising in the absence of distance restraints (panels A and B) are evident. Instead of undergoing the change necessary to switch between the cation and anion complexing conformation, the ion is just expelled from
Figure 6.2: Configurations of the cyclic aminooxy-hexapeptide complexed with Cl\(^-\) (A) and with Na\(^+\) (B) from MD simulations in (explicit) chloroform.

the peptide. On typical MD simulation time-scales, the ion will not find its way back to form a stable complex. Therefore, without distance restraints, no reliable free energy difference can be obtained. The hidden restraints could be successfully applied to stabilize an otherwise unusable transition path and to calculate a free energy difference between the two states (panels C and D).

<table>
<thead>
<tr>
<th>process</th>
<th>hidden restraints</th>
<th>unrestrained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cl(^-) → Na(^+)</td>
<td>-149 ±16</td>
<td>134 ±10</td>
</tr>
<tr>
<td>Na(^+) → Cl(^-)</td>
<td>150 ±13</td>
<td>245 ±11</td>
</tr>
</tbody>
</table>

Table 6.1: Numerically integrated free energy difference ΔG, in kJ/mol, obtained by multi-configurational thermodynamic integration of the cyclic aminooxy-hexapeptide ion complex in chloroform, where the ion was changed from Cl\(^-\) to Na\(^+\) and back. Averages \(\langle \frac{\partial H}{\partial \lambda} \rangle\) were obtained from 250 ps of simulations (after 50 ps of equilibration) at 11 discrete λ values. Error estimates result from block averaging and extrapolating the block length to infinity\(^40\).

6.4.2 Relative stabilities of hexopyranose in \(^4C_1\) vs. \(^1C_4\) conformation

A new GROMOS force-field parameter set, 45A4, has become available for explicit solvent (water) simulations of hexopyranose-based carbohydrates\(^41\). It was obtained according to the fol-
Figure 6.3: Average derivative of the Hamiltonian with respect to the (alchemical) pathway parameter $\lambda$, obtained from 250 ps simulations (after 50 ps equilibration) at 11 discrete $\lambda$ values for the cyclic aminoxyl-hexapeptide ion complex in chloroform. In the upper half (panels A and B) results from unrestrained simulations are shown, the lower half (panels C and D) represents simulations including hidden restraints. On the left side (panels A and C) thermodynamic integration along $\lambda$ representing a change from $\text{Cl}^-$ ($\lambda = 0$) to $\text{Na}^+$ ($\lambda = 1$) was carried out, whereas the right side (panels B and D) corresponds to the reverse transformation from $\text{Na}^+$ ($\lambda = 0$) to $\text{Cl}^-$ ($\lambda = 1$).

Following procedure: (1) reassigning the atomic partial charges based on a fit to the quantum-mechanical electrostatic potential around a trisaccharide; (2) refining the torsional potential energy function parameters associated with the rotations of the hydroxymethyl, hydroxyl, and anomic alkoxy groups by fitting to corresponding quantum-mechanical profiles for hexopyranosides; (3) adapting the rotational potential energy function parameters determining the ring conformation so as to stabilize the (experimentally predominant) $^4C_1$ chair conformation (Figure 6.5).

In unrestrained simulations, starting from the $^1C_4$ conformation, isomerisation to the dominant $^4C_1$ conformation occurred within at most 2 ns. Starting from the $^4C_1$ conformation, no isomerisation occurred during 5 ns\textsuperscript{41}. To calculate the free energy difference between the two
Figure 6.4: Average distances between the ion and the six carbonyl oxygen atoms (in red) and the six amide hydrogen atoms (in black) of the cyclic aminooxy hexapeptide ion complex are shown, obtained from 250 ps simulation (after 50 ps equilibration) at 11 discrete \( \lambda \) values during a multi-configurational thermodynamic integration from \( \text{Cl}^- (\lambda = 0) \) to \( \text{Na}^+ (\lambda = 1) \) (left side, panels A and C) and backwards from \( \text{Na}^+ (\lambda = 0) \) to \( \text{Cl}^- (\lambda = 1) \) (right side, panels B and D), without restraints (panels A and B) and with (hidden) restraints (panels C and D).

conformations hidden (dihedral-angle) restraints were applied to force a transition from \( ^4\text{C}_1 \) to \( ^1\text{C}_4 \). The restraints were applied to the ring dihedral angles, with reference values \( \phi_{\text{A}}^0 \) and \( \phi_{\text{B}}^0 \) in the two end states \( ^4\text{C}_1 \) and \( ^1\text{C}_4 \) according to Table 6.2. The atom numbering is specified in Figure 6.6. During the isomerization, the dihedral angles will go through the syn conformation (torsional dihedral angles of 0°). Therefore, the maximum value of the restraining potential energy function was set to be reached at \( \phi_{k}^{\text{max}} = -180^\circ \). 200 ps of MD simulations (including first 50 ps of equilibration) were performed in explicit SPC water\(^{42} \), at 11 discrete \( \lambda \) values (\( \lambda = 0.0, 0.1, 0.2, ..., 1.0 \)). The exponents \( m \) and \( n \) of the hidden restraints in Equation 6.1 were chosen as \( m = 2 \) and \( n = 2 \), and the force constants \( K_{\text{dih}} = 10\text{kJ}/\text{rad}^2 \) with all six individual weight factors equal to 1.0. A temperature of 300 K (separate coupling of solute and solvent, \( \tau_T = 0.1 \text{ps} \)) and pressure of 1 atm (\( \tau_P = 0.5 \text{ps} \), \( \kappa_P = 4.575 \times 10^{-4} \text{(kJ mol}^{-1}\text{nm}^{-3})^{-1} \) using isotropic scaling of coordinates) were maintained by weak coupling\(^{33} \), bond lengths were constrained using the SHAKE alg...
Table 6.2: Values (in degrees) of the six dihedral angles of the sugar ring in β-D-glucopyranoside for the $^4C_1$ and the $^1C_4$ conformations. The atom names are specified in Figure 6.6.
Figure 6.6: Atom numbering of the β-D-glucopyranoside.
values and the corresponding derivative of the Hamiltonian is depicted in Figure 6.8.

Figure 6.7: Average values of the ring dihedral angles (see Table 6.2) during a multi-configurational thermodynamic integration of $\beta$-D-glucopyranoside in explicit SPC water with 150 ps simulation (and 50 ps equilibration) at each discrete $\lambda$ point. The dihedral-angle averages for a change from the $^4C_1 (\lambda = 0)$ to the $^1C_4 (\lambda = 1)$ conformation are indicated by circles, connected with dotted lines, the ones for the backward change from the $^1C_4$ to the $^4C_1$ conformation are indicated by crosses. The order of the dihedral-angles as given in Table 6.2 corresponds to the colors black, red, green, blue, yellow and indigo.

Using thermodynamic integration (Equation 6.2), the free energy difference between the two conformations is calculated to be $4.6 \pm 3.4$ kJ/mol for the change from $^4C_1$ to $^1C_4$ and $-5.6 \pm 2.5$ kJ/mol backwards. The (internal) potential energy difference between the two conformations had been calculated to be $35$ kJ/mol\textsuperscript{41}, so interactions with the solvent and entropy are lowering the difference significantly.

For comparison, the potential of mean force of the transition from the $^4C_1$ to the $^1C_4$ state of the $\beta$-D-glucopyranoside was also calculated using dihedral-angle constraints (Figure 6.9). The ring dihedral angles (see Table 6.2) were changed linearly with the pathway parameter $\lambda$ and 51 discrete $\lambda$-points were used. At each $\lambda$ point a short equilibration (5 ps) was followed by (at
Figure 6.8: Average derivative of the Hamiltonian with respect to the pathway coordinate $\lambda$ during a multi-configurational thermodynamic integration of $\beta$-D-glucopyranoside in explicit SPC water with 150 ps simulation (after 50 ps equilibration) at each discrete $\lambda$ point. The dashed line with bold error bars corresponds to a conformational change from $^4C_1$ ($\lambda = 0$) to $^1C_4$ ($\lambda = 1$), the dotted line and thin error bars to the reverse isomerization.

At least 50 ps of simulation over which the contributions of the constraint forces to the potential of mean force (see Appendix, Equation 6.59) were averaged. The iterative procedure applied to solve the coupled equations with terms up to eighth power in $\lambda$ generally converges well, even in case the dihedral constraints form a cycle (in the sugar ring). But for specific values of the dihedral angle ($90^0, 180^0, 270^0$ and $360^0$), the linearized equations for the Lagrange multipliers (Equation 6.58) yield $l_k$'s that are close to zero. In those cases, the iterative procedure to solve for the constraint forces does not converge. Therefore, no mean force could be calculated for $\lambda$-values 0.48, 0.50 and 0.52. Integrating the potential of mean force leads to a relative free energy difference between the $^4C_1$ and the $^1C_4$ state of 8.6 ± 10.2 kJ/mol. Almost twice the amount of simulation time was spent to get this potential of mean force compared to the simulations using dihedral-angle restraints. Yet, the curve is quite rough and the uncertainty is quite high. Therefore it is not possible to estimate the influence the constraining of the end-state has on the
Figure 6.9: Potential of mean force with respect to the pathway coordinate $\lambda$ of $\beta$-D-glucopyranoside ($\lambda = 0$: $^4C_1$ and $\lambda = 1$: $^1C_4$) in explicit SPC water with constrained dihedral angles (see Table 6.2), with (at least) 50 ps simulation (after 5 ps equilibration) at each discrete $\lambda$ point (in total 48 points). No mean force could be calculated for $\lambda$-values of 0.48, 0.50 and 0.52 because the Lagrange multipliers $l_k$ of the dihedral-angle constraints approach zero and the iterative solution of Equation 6.58 does not converge.

6.5 Discussion

Calculating relative free energies between pairs of different (meta)stable states may be quite demanding. On the one hand, the two states may be separated by high potential energy barriers which have to be overcome. On the other hand, very extensive sampling may be necessary to have sufficient data on both states. Both problems can be mediated by applying restraints. But at the same time, the end states of the thermodynamic integration may be influenced by the restraints, making the free energy difference between the end states dependent on the path chosen.
This pathway dependence can be avoided by the use of hidden restraints which are characterized by zero restraining energy and forces in the end states. Using hidden distance restraints, the relative free energy difference of changing from an anion to a cation in a cyclic D,L-α-aminoxy-hexapeptide-ion complex, accompanied by large structural changes in the peptide, could be calculated by guiding the simulation along a transition path. Application of hidden dihedral-angle restraints enforced the isomerization of a β-D-glucopyranoside from a $^4C_1$ to a $^1C_4$ conformation and backwards. The former conformational change (from the more to the less stable structure), was not observed during 5 ns of unrestrained MD simulations.

Additionally, harmonic dihedral-angle restraints have been enhanced by an additional parameter $\phi_k^\text{max}$ to define the angle with maximum restraint potential energy. Using this parameter, the direction of rotation around the dihedral angle can be controlled.

In contrast to standard potential of mean-force calculations, hidden restraints allow relative free energy calculations of unrestrained end states, which makes the results independent of the pathway chosen. As the restraints are only used to guide the simulation and most of the time only with weak influence, the simulated system retains a certain flexibility around the specified transition path. On the other hand, if a steep barrier must be crossed, the system will first lag behind the pathway parameter $\lambda$, then suddenly cross the barrier and catch up. During the leap over the barrier, no equilibrium (average) derivative of the Hamiltonian with respect to $\lambda$ can be obtained. This can be seen by peaks or jumps in the free energy derivative and by large root-mean-square deviations (or error estimates) of its average values at discrete $\lambda$ points. This problem may be alleviated by lowering the high barriers that separate the two states during the thermodynamic integration, if the force-field terms contributing to these energy barriers can be identified.

The potential of mean force obtained using dihedral-angle constraints of the transition of β-D-glucopyranoside from the state $^4C_1$ to $^1C_4$ was compared to the $\lambda$-derivative of the Hamiltonian in multi-configurational thermodynamic integration using hidden restraints. Using hidden restraints was straightforward, whereas dihedral-angle constraining turned out to be difficult because the iterative procedure to obtain the Lagrange multipliers that determine the contributions of the constraint forces to the potential of mean force does not converge for all dihedral-angle values. Also, even though more simulation time was spent, the error estimates are considerably larger. We conclude that use of hidden restraints constitutes an efficient means to obtain free energy differences between states that are rarely sampled in unrestrained simulations.

6.6 Appendix

A potential of mean force can alternatively be obtained by constraining the system to a particular $\lambda$-value, calculating the derivative of the free energy with respect to $\lambda$ ($dF/d\lambda$) from the constraint forces ($f_c$) and repeat this for the range of $\lambda$-values connecting states A and B of the
system. Below we present expressions for $dF/d\lambda$ and $f^c$ when applying distance constraints or dihedral-angle constraints.

### 6.6.1 Distance constraints

A set of $k = 1, 2, \ldots, N_c$ distance constraints between atoms with positions $r_{k1}$ and $r_{k2}$ can be written as

$$\sigma_k(r; r_{k1}^0(\lambda)) \equiv r_{k1k2}^2 - (r_{k1k2}^0(\lambda))^2 = 0, \quad k = 1, 2, \ldots, N_c$$

(6.20)

where $r_{k1k2} \equiv r_{k1} - r_{k2}$, $k \equiv (k_1, k_2)$, and the distance

$$r_{k1k2} \equiv |r_{k1k2}| \equiv \left( (x_{k1} - x_{k2})^2 + (y_{k1} - y_{k2})^2 + (z_{k1} - z_{k2})^2 \right)^{\frac{1}{2}},$$

(6.21)

is constrained to the $\lambda$-dependent value

$$r_{k1k2}^0(\lambda) = (1 - \lambda)r_{k1k2}^0A + \lambda r_{k1k2}^0B,$$

(6.22)

in which $r_{k1k2}^0A$ is the distance constraint value in state $A$ and $r_{k1k2}^0B$ that in state $B$. Furthermore, we use the notation $r = (r_1, r_2, \ldots, r_N)$ for a configuration of $N$ atoms. The use of $r_{k1k2}^2$ to define the constraint $\sigma_k$ in Equation 6.20 instead of $r_{k1k2}$ leads to simpler equations to obtain the constraint forces and their contribution to the potential of mean force.

Newton’s equations of motion for $N$ atoms with masses $m_i$ including a potential energy function $\mathcal{V}(r)$ and the constraints $\sigma_k$ multiplied with the Lagrange multipliers $l_k(t)$ are

$$m_i \frac{d^2 r_i(t)}{dt^2} = -\frac{\partial}{\partial r_i} \left( \mathcal{V}(r) + \sum_{k=1}^{N_c} l_k(t) \sigma_k(r; r_{k1k2}^0(\lambda)) \right), \quad i = 1, 2, \ldots, N.$$ 

(6.23)

The Lagrange multipliers $l_k(t)$ are to be determined such that the condition given in Equation 6.20 is satisfied. The first term on the right in Equation 6.23 represents the unconstrained force $f_i^c(t)$ derived from the interaction function $\mathcal{V}(r)$ and the second term represents the (yet unknown) constraint force $f_i^c(t)$,

$$f_i^c(t) = -\sum_{k=1}^{N_c} l_k(t) \frac{\partial \sigma_k(r; r_{k1k2}^0(\lambda))}{\partial r_i}$$

(6.24)

$$= -2 \sum_{k=1}^{N_c} (\delta_{ik1} - \delta_{ik2}) l_k(t) r_{k1k2}(t).$$

The leap-frog scheme to integrate Newton’s equations of motion using a timestep $\Delta t$ yields for the unconstrained positions at time $t_n + \Delta t$,

$$r_i^{mc}(t_n + \Delta t) = r_i(t_n) + v_i(t_n - \Delta t/2)\Delta t + m_i^{-1} f_i^c(t_n)(\Delta t)^2,$$ 

(6.25)
where the atomic velocities are indicated by \( v_i \). The constrained positions at time \( t_n + \Delta t \) are related to the constraint forces through

\[
\mathbf{r}_i(t_n + \Delta t) = \mathbf{r}_{i}^{\text{nc}}(t_n + \Delta t) + m_i^{-1} \mathbf{F}_i(t_n)(\Delta t)^2,
\]

and should satisfy the constraint Equation 6.20,

\[
\sigma_k(\mathbf{r}(t_n + \Delta t); r_{k,k'}^0(\lambda)) = 0, \quad k = 1, 2, \ldots, N_c
\]

which yields the following equations for the Lagrange multipliers \( l_k(t_n) \),

\[
\begin{bmatrix}
\mathbf{r}_{k_1}^{nc}(t_n + \Delta t) - 2m_{k_1}^{-1}(\Delta t)^2 \sum_{k'=1}^{N_c} (\delta_{k_1,k'} - \delta_{k_1,k''}) l_k(t_n) \mathbf{r}_{k',k''}(t_n) \\
- \mathbf{r}_{k_2}^{nc}(t_n + \Delta t) + 2m_{k_2}^{-1}(\Delta t)^2 \sum_{k'=1}^{N_c} (\delta_{k_2,k'} - \delta_{k_2,k''}) l_k(t_n) \mathbf{r}_{k',k''}(t_n) \\
- \left[(1 - \lambda) r_{k_1,k_2}^{0A} + \lambda r_{k_1,k_2}^{0B}\right]^2 = 0
\end{bmatrix}
\]

This is a set of \( N_c \) quadratic equations in the \( N_c \) unknowns \( l_k(t_n) \). It can be solved by linearization (neglect of terms quadratic in \( l_k(t_n) \)) followed by matrix inversion or by sequentially solving the linearized equations for each constraint omitting the coupling between the different constraints (equations), and iterating through all the equations until the \( l_k(t_n) \) converge to a consistent value. The latter method is used in the procedure SHAKE\textsuperscript{34}. The quadratic, decoupled equation for the Lagrange multiplier \( l_k(t_n) \) is

\[
\begin{bmatrix}
\mathbf{r}_{k_1,k_2}^{nc}(t_n + \Delta t) - l_k(t_n)(\Delta t)^2 (m_{k_1}^{-1} + m_{k_2}^{-1}) \mathbf{r}_{k_1,k_2}(t_n) \\
- \left[(1 - \lambda) r_{k_1,k_2}^{0A} + \lambda r_{k_1,k_2}^{0B}\right]^2 = 0
\end{bmatrix}
\]

After linearization one finds

\[
l_k(t_n) = \frac{\left[(1 - \lambda) r_{k_1,k_2}^{0A} + \lambda r_{k_1,k_2}^{0B}\right]^2 - \left[\mathbf{r}_{k_1,k_2}^{nc}(t_n + \Delta t)\right]^2}{-4(\Delta t)^2 (m_{k_1}^{-1} + m_{k_2}^{-1}) \mathbf{r}_{k_1,k_2}(t_n) \cdot \mathbf{r}_{k_1,k_2}^{nc}(t_n + \Delta t)}.
\]

the Lagrange multipliers at timestep \( t_n \).

The derivative of the free energy \( F(\lambda) \) with respect to \( \lambda \) for a system including distance constraints is\textsuperscript{5}

\[
\frac{dF}{d\lambda} = < \frac{\partial K}{\partial \lambda} >_\lambda + < \frac{\partial \mathcal{V}}{\partial \lambda} >_\lambda + < \frac{\partial}{\partial \lambda} \sum_{k=1}^{N_c} l_k \sigma_k(\mathbf{r}; r_{k,k'}^0(\lambda)) >_\lambda.
\]

The symbol \(< ... >_\lambda \) denotes an ensemble average over the constrained simulation at the value \( \lambda \). The first term on the right contains the possible contribution of the kinetic energy \( K(p,r) \), the
second the contribution of the unconstrained interaction terms and the third the contribution of the constraint forces, which can be expressed for the k-th constraint as

\[
\frac{dF_k^c(\lambda)}{d\lambda} = -2 < l_k >^0_{\text{k1k2}}(\lambda) \left( \frac{0.0,0A}{r_{\text{k1k2}}} - \frac{0.0A}{r_{\text{k1k2}}} \right).
\] (6.32)

The total contribution of the \(N_c\) constraints to the potential of mean force is then

\[
\frac{dF^c(\lambda)}{d\lambda} = \sum_{k=1}^{N_c} \frac{dF_k^c(\lambda)}{d\lambda}.
\] (6.33)

### 6.6.2 Dihedral-angle constraints

For dihedral-angle constraints, the derivation of the expressions for the constraint forces \(F^c\) and their contribution \(dF^c/d\lambda\) to the free energy \(F(\lambda)\) follows the same lines as that for the distance constraints. However, due to the not very simple dependence of a dihedral angle \(\phi_k(\mathbf{r})\) upon the positions \(\mathbf{r}_{\text{k1}}, \mathbf{r}_{\text{k2}}, \mathbf{r}_{\text{k3}}\) and \(\mathbf{r}_{\text{k4}}\) of its four constituting atoms \(k_1, k_2, k_3\) and \(k_4\) (i.e. \(k_1 - k_2 - k_3 - k_4\)), the formulae become much more complicated.

Expressions for \(\phi_k(\mathbf{r}_{\text{k1}}, \mathbf{r}_{\text{k2}}, \mathbf{r}_{\text{k3}}, \mathbf{r}_{\text{k4}})\) can be found in the literature\textsuperscript{15-17},

\[
\phi_k = \text{sign}(\phi_k)\arccos \left( \frac{\mathbf{r}_{\text{k5k2}} \cdot \mathbf{r}_{\text{k6k3}}}{|\mathbf{r}_{\text{k5k2}}||\mathbf{r}_{\text{k6k3}}|} \right),
\] (6.34)

where

\[
\mathbf{r}_{\text{k5k2}} \equiv \mathbf{r}_{\text{k1k2}} \times \mathbf{r}_{\text{k3k2}},
\] (6.35)

\[
\mathbf{r}_{\text{k6k3}} \equiv \mathbf{r}_{\text{k1k2}} \times \mathbf{r}_{\text{k3k4}},
\] (6.36)

and

\[
\text{sign}(\phi_k) = \text{sign}(\mathbf{r}_{\text{k1k2}} \cdot \mathbf{r}_{\text{k6k3}}),
\] (6.37)

following the IUPAC-IUB convention\textsuperscript{48}. Since \(0 \leq \arccos \leq \pi\), we have

\[-\pi \leq \phi_k \leq \pi.
\] (6.38)

Because of the occurrence of the \(\arccos\) function in the definition of the dihedral angle \(\phi_k\) the constraints are to be formulated in terms of \(\cos(\phi_k)\). The occurrence of the square root functions in the distances \(|\mathbf{r}_{\text{k5k2}}|\) and \(|\mathbf{r}_{\text{k6k3}}|\) in the denominator of Equation 6.34 suggests that the use of \(\cos^2(\phi_k)\) will simplify the expressions. Thus, we consider a set of \(N_c\) dihedral-angle constraints

\[
\sigma_k(\phi_k(\mathbf{r}); \phi_k^0(\lambda)) \equiv \cos^2(\phi_k(\mathbf{r})) - \cos^2(\phi_k^0(\lambda)) = 0, \quad k = 1, 2, \ldots, N_c
\] (6.39)

where the angle \(\phi_k(\mathbf{r})\) is constrained to the \(\lambda\)-dependent value

\[
\phi_k^0(\lambda) = (1 - \lambda)\phi_k^{0,0} + \lambda \phi_k^{0,0A},
\] (6.40)
in which $\phi_k^{A}$ is the $\phi_k$-value in state $A$ and $\phi_k^{B}$ that in state $B$.

Newton’s equations of motion for $N$ atoms become

$$\frac{d^2 \mathbf{r}_i(t)}{dt^2} = -\frac{\partial}{\partial \mathbf{r}_i} \left( \mathcal{V}'(\mathbf{r}) + \sum_{k=1}^{N_c} l_k(t) \sigma_k(\phi_k(\mathbf{r}); \phi_k^0(\lambda)) \right), \quad i = 1, 2, \ldots, N$$

(6.41)

where the Lagrange multipliers $l_k(t)$ are to be determined such that the condition given in Equation 6.39 is satisfied. The second term on the right in Equation 6.41 represents the (yet unknown) constraint forces,

$$f_i^c(t) = \sum_{k=1}^{N_c} l_k(t) \frac{\partial \sigma_k(\phi_k(\mathbf{r}); \phi_k^0(\lambda))}{\partial \mathbf{r}_i} + \sum_{k=1}^{N_c} l_k(t) 2\cos(\phi_k) \sin(\phi_k) \frac{\partial \phi_k(\mathbf{r})}{\partial \mathbf{r}_i}.$$  

(6.42)

Expressions for $\frac{\partial \phi_k}{\partial \mathbf{r}_i}$ can be found in the literature too15–17,

$$\frac{\partial \phi_k(\mathbf{r})}{\partial \mathbf{r}_i} = \delta_{ik_1} \frac{|\mathbf{r}_{k_1k_2}|}{|\mathbf{r}_{k_1k_2}|^2 \mathbf{r}_{k_1k_2}} + \delta_{ik_2} \left[ \left( \frac{|\mathbf{r}_{k_1k_2} \cdot \mathbf{r}_{k_3k_2}|}{|\mathbf{r}_{k_3k_2}|^2 |\mathbf{r}_{k_1k_2}|} - 1 \right) \frac{|\mathbf{r}_{k_1k_2}|^2 \mathbf{r}_{k_3k_2}}{|\mathbf{r}_{k_1k_2}|^2 |\mathbf{r}_{k_3k_2}|^2 \mathbf{r}_{k_1k_2}} + \frac{|\mathbf{r}_{k_1k_2}|}{|\mathbf{r}_{k_3k_2}|^2 |\mathbf{r}_{k_1k_2}|^2 \mathbf{r}_{k_3k_2}} \right]$$

(6.43)

To shorten the expressions we denote the four terms in Equation 6.43, apart from the Kronecker delta’s, by $a_{k_1}$, $a_{k_2}$, $a_{k_3}$, and $a_{k_4}$, respectively. Then we have

$$f_i^c(t) = \sum_{k=1}^{N_c} l_k(t) \sin(2\phi_k(t)) \left[ \delta_{ik_1} a_{k_1}(t) + \delta_{ik_2} a_{k_2}(t) + \delta_{ik_3} a_{k_3}(t) + \delta_{ik_4} a_{k_4}(t) \right].$$

(6.44)

The leap-frog scheme yields the unconstrained positions $r_i^{uc}(t_n + \Delta t)$ from Equation 6.25. The constrained positions $r_i(t_n + \Delta t)$ are related to the constraint forces (Equation 6.44) through Equation 6.26 and should satisfy the constraint Equations 6.39,

$$\cos^2(\phi_k(\mathbf{r}(t_n + \Delta t))) - \cos^2(\phi_k^0(\lambda)) = 0, \quad k = 1, 2, \ldots, N_c,$$

(6.45)

or using Equation 6.34,

$$\left[ \frac{|\mathbf{r}_{k_1k_2}(t_n + \Delta t) \cdot \mathbf{r}_{k_1k_2}(t_n + \Delta t)|^2}{|\mathbf{r}_{k_1k_2}(t_n + \Delta t)||\mathbf{r}_{k_1k_2}(t_n + \Delta t)|} \right] - \cos^2(\phi_k^0(\lambda)) = 0.$$  

(6.46)
Chapter 6. Sampling of rare events using hidden restraints

Since $r_{k,k_2}(t_n + \Delta t)$ and $r_{k,k_3}(t_n + \Delta t)$ are each quadratic in the Lagrange multipliers $l_k(t_n)$, both the numerator and the denominator of the left term in Equation 6.46 contain powers of up to eight of the $l_k(t_n)$. Thus a set of $N_c$ equations consisting of terms containing up to powers of eight of the unknowns $l_k(t_n)$ is to be solved. As for the case of distance constraints this is achieved by linearizing the equations for each constraint, omitting the coupling between the different constraints (equations), and iterating through all $N_c$ equations until the $l_k(t_n)$ converge to a consistent value.

Using Equations 6.26 and 6.44 we find for the $k$-th constraint

$$r_{k_1k_2}(t_n + \Delta t) = r_{k_1k_2}^{\mu c}(t_n + \Delta t) + \frac{l_k(t_n) \sin(2\phi_k(t_n))}{(\Delta t)^2} \left( m_{k_1}^{-1} a_{k_1}(t_n) - m_{k_2}^{-1} a_{k_2}(t_n) \right)$$

and likewise for $r_{k_3k_2}(t_n + \Delta t)$ and $r_{k_3k_4}(t_n + \Delta t)$. Building the cross products in Equations 6.35 and 6.36 and linearizing the resulting expressions yields

$$r_{k_3k_2}(t_n + \Delta t) = r_{k_3k_2}^{\mu c}(t_n + \Delta t) \times r_{k_3k_2}^{\mu c}(t_n + \Delta t) + \frac{l_k(t_n) \sin(2\phi_k(t_n))}{(\Delta t)^2} \left[ r_{k_1k_2}^{\mu c}(t_n + \Delta t) \times \left( m_{k_3}^{-1} a_{k_3}(t_n) - m_{k_2}^{-1} a_{k_2}(t_n) \right) 
- r_{k_3k_2}^{\mu c}(t_n + \Delta t) \times \left( m_{k_1}^{-1} a_{k_1}(t_n) - m_{k_2}^{-1} a_{k_2}(t_n) \right) \right]$$

or using a shorter notation $b_{k_1k_2k_3}(t_n + \Delta t)$ for the last factor

$$r_{k_3k_2}(t_n + \Delta t) = r_{k_3k_2}^{\mu c}(t_n + \Delta t) \times r_{k_3k_2}^{\mu c}(t_n + \Delta t) + \frac{l_k(t_n) \sin(2\phi_k(t_n))}{(\Delta t)^2} b_{k_3k_2k_3}(t_n, t_n + \Delta t),$$

and

$$r_{k_3k_4}(t_n + \Delta t) = r_{k_3k_2}^{\mu c}(t_n + \Delta t) \times r_{k_3k_4}^{\mu c}(t_n + \Delta t) + \frac{l_k(t_n) \sin(2\phi_k(t_n))}{(\Delta t)^2} \left[ r_{k_3k_2}^{\mu c}(t_n + \Delta t) \times \left( m_{k_3}^{-1} a_{k_3}(t_n) - m_{k_4}^{-1} a_{k_4}(t_n) \right) 
- r_{k_3k_4}^{\mu c}(t_n + \Delta t) \times \left( m_{k_3}^{-1} a_{k_3}(t_n) - m_{k_2}^{-1} a_{k_2}(t_n) \right) \right]$$

or using the shorter notation

$$r_{k_3k_4}(t_n + \Delta t) = r_{k_3k_4}^{\mu c}(t_n + \Delta t) \times r_{k_3k_4}^{\mu c}(t_n + \Delta t) + \frac{l_k(t_n) \sin(2\phi_k(t_n))}{(\Delta t)^2} b_{k_3k_4}(t_n, t_n + \Delta t).$$
The scalar product in the numerator of the first term in Equation 6.46 becomes after linearization

\[
r_{k3k2}(t_n + \Delta t) \cdot r_{k3k3}(t_n + \Delta t) =
\begin{align*}
&\left( r_{k1k2}^{uc}(t_n + \Delta t) \times r_{k3k2}^{uc}(t_n + \Delta t) \right) \cdot \left( r_{k3k2}^{uc}(t_n + \Delta t) \times r_{k3k3}^{uc}(t_n + \Delta t) \right) \\
&+ l_k(t_n) \sin(2\phi_k(t_n))(\Delta t)^2 \\
&\left[ \left( r_{k1k2}^{uc}(t_n + \Delta t) \times r_{k3k2}^{uc}(t_n + \Delta t) \right) \cdot b_{k2k3k4}(t_n, t_n + \Delta t) + \\
&\left( r_{k1k2}^{uc}(t_n + \Delta t) \times r_{k3k4}^{uc}(t_n + \Delta t) \right) \cdot b_{k1k3k3}(t_n, t_n + \Delta t) \right]
\end{align*}
\] (6.52)

or in a shorter notation

\[
r_{k3k2}(t_n + \Delta t) \cdot r_{k3k3}(t_n + \Delta t) = c_{k1k2k3k4}(t_n + \Delta t) \\
+ l_k(t_n) \sin(2\phi_k(t_n))(\Delta t) d_{k1k2k3k4}(t_n, t_n + \Delta t).
\] (6.53)

The square becomes after linearization

\[
\left( r_{k3k2}(t_n + \Delta t) \cdot r_{k3k3}(t_n + \Delta t) \right)^2 = (c_{k1k2k3k4}(t_n + \Delta t))^2 + 2l_k(t_n) \sin(2\phi_k(t_n))(\Delta t)^2 c_{k1k2k3k4}(t_n + \Delta t)d_{k1k2k3k4}(t_n, t_n + \Delta t).
\] (6.54)

The factors in the denominator of the first term in Equation 6.46 become

\[
|r_{k3k2}(t_n + \Delta t)|^2 = \left( r_{k1k2}^{uc}(t_n + \Delta t) \times r_{k3k2}^{uc}(t_n + \Delta t) \right)^2 + 2l_k(t_n) \sin(2\phi_k(t_n))(\Delta t)^2 \\
\left( r_{k1k2}^{uc}(t_n + \Delta t) \times r_{k3k2}^{uc}(t_n + \Delta t) \right) \cdot b_{k1k2k3}(t_n + \Delta t)
\] (6.55)

and

\[
|r_{k3k3}(t_n + \Delta t)|^2 = \left( r_{k3k2}^{uc}(t_n + \Delta t) \times r_{k3k4}^{uc}(t_n + \Delta t) \right)^2 + 2l_k(t_n) \sin(2\phi_k(t_n))(\Delta t)^2 \\
\left( r_{k3k2}^{uc}(t_n + \Delta t) \times r_{k3k4}^{uc}(t_n + \Delta t) \right) \cdot b_{k2k3k4}(t_n + \Delta t).
\] (6.56)

The linearized denominator of the first term in Equation 6.46 is then

\[
|r_{k3k2}(t_n + \Delta t)|^2 |r_{k3k3}(t_n + \Delta t)|^2 =
\begin{align*}
&\left( r_{k1k2}^{uc}(t_n + \Delta t) \times r_{k3k2}^{uc}(t_n + \Delta t) \right)^2 \left( r_{k3k2}^{uc}(t_n + \Delta t) \times r_{k3k4}^{uc}(t_n + \Delta t) \right)^2 \\
&+ 2l_k(t_n) \sin(2\phi_k(t_n))(\Delta t)^2 \\
&\left[ \left( r_{k1k2}^{uc}(t_n + \Delta t) \times r_{k3k2}^{uc}(t_n + \Delta t) \right)^2 \\
&\left( r_{k4k2}^{uc}(t_n + \Delta t) \times r_{k3k4}^{uc}(t_n + \Delta t) \right) \cdot b_{k2k3k4}(t_n, t_n + \Delta t) + \\
&\left( r_{k3k2}^{uc}(t_n + \Delta t) \times r_{k3k4}^{uc}(t_n + \Delta t) \right)^2 \right]
\end{align*}
\] (6.57)
Finally, the equation for the Lagrange multiplier of the k-th constraint becomes

$$l_k(t_n) = \left[ \cos^2(\phi_k^0(\lambda)) \left( \mathbf{r}_{k_1k_2}^{uc}(t_n + \Delta t) \times \mathbf{r}_{k_3k_4}^{uc}(t_n + \Delta t) \right)^2 \right]$$

$$+ \frac{\sin(2\phi_k(t_n))(\Delta t)^2 c_{k_1k_2k_3k_4}(t_n + \Delta t) d_{k_1k_3k_3k_4}(t_n + \Delta t)}{\cos^2(\phi_k^0(\lambda))}$$

$$- \left[ \left( \mathbf{r}_{k_1k_2}^{uc}(t_n + \Delta t) \times \mathbf{r}_{k_3k_4}^{uc}(t_n + \Delta t) \right)^2 \mathbf{b}_{k_2k_3k_4}(t_n, t_n + \Delta t) \right]^{-1} \left( \mathbf{r}_{k_1k_3}^{uc}(t_n + \Delta t) \times \mathbf{r}_{k_2k_4}^{uc}(t_n + \Delta t) \right) \mathbf{b}_{k_1k_3k_3}(t_n, t_n + \Delta t) \quad (6.58)$$

The derivative of the contribution of the constraint forces to the free energy for the k-th constraint becomes

$$\frac{dF^C(\lambda)}{d\lambda} = \langle l_k \rangle \sin(2\phi_k^0(\lambda)) \left( \phi_k^0 - \phi_k^{0,A} \right) \quad (6.59)$$

We note that the expressions given in this Appendix for the application of dihedral-angle constraints are different from the formalism presented in\(^{49}\), which is based on matrix inversion.

### 6.7 Acknowledgments

Financial support by the National Center of Competence in Research (NCCR) Structural Biology of the Swiss National Science Foundation (SNSF) is gratefully acknowledged.

### 6.8 Bibliography


6.8. Bibliography


Chapter 7

Adaptive restraints using local-elevation simulation

7.1 Summary

Introducing experimental values as restraints into molecular dynamics (MD) simulation to bias the values of particular molecular properties, such as nuclear Overhauser effect intensities or distances, dipolar couplings, $^3J$-coupling constants, chemical shifts or crystallographic structure factors, towards experimental values is a widely used structure refinement method. Because multiple torsion angle values $\phi$ correspond to the same $^3J$-coupling constant and high-energy barriers are separating those, restraining $^3J$-coupling constants remains difficult. A method to adaptively enforce restraints using a local elevation (LE) potential energy function is presented and applied to $^3J$-coupling constant restraining in an MD simulation of hen egg-white lysozyme (HEWL). The method successfully enhances sampling of the restrained torsion angles until the 37 experimental $^3J$-coupling constant values are reached, thereby also improving the agreement with the 1630 experimental NOE atom-atom distance upper bounds. Afterwards the torsional angles $\phi$ are kept restrained by the built-up local-elevation potential energies.
Chapter 7. Adaptive restraints using local-elevation simulation

7.2 Introduction

Experimental techniques such as X-ray diffraction and NMR spectroscopy are widely used to derive structural information from molecules in solution, solid state or in crystal form. These experimental methods have in common that the values of observable quantities are averages over time and over an ensemble of molecules. It may even not be possible to come up with a single physically plausible structure or conformation reproducing all experimental values\(^1,2\). Therefore, the corresponding properties of an MD simulation should be calculated as time averages and when restraints are applied, those should reproduce the experimental values on average\(^3\). The latter can be achieved by adding a penalty function \(V_{\text{restr}}\) to the physical force field \(V_{\text{phys}}\) of the MD simulation\(^4\),

\[
V(\mathbf{r}(t)) = V_{\text{phys}}(\mathbf{r}(t)) + V_{\text{restr}}(\mathbf{r}(t)),
\]

using a penalty function \(V_{\text{restr}}\) of the form\(^5\)

\[
V_{\text{restr}}(\mathbf{r}(t)) = \sum_{k=1}^{N_{\text{restr}}} \frac{1}{2K_k} (q_k(\mathbf{r}(t)) - q_k^0)^2 (q_k(\mathbf{r}(t)) - q_k^0)^2,
\]

where \(q(\mathbf{r}(t))\) may be a weighted average during the simulation\(^3\) and \(q(\mathbf{r}(t))\) is any of the above mentioned observables\(^6\). By introduction of the first quadratic factor of Equation 7.2, the functional form given here avoids generating large artificial structural fluctuations, as observed when using standard time-averaging \(^3\)-value restraints\(^7,8\).

The \(^3\)-coupling constants are usually calculated using the Karplus relation\(^9\)

\[
J(\theta(\mathbf{r}(t))) = a\cos^2\theta + b\cos\theta + c,
\]

where \(\theta\) is the torsion angle defined by the four covalently bound atoms that determine a particular \(^3\)-coupling constant. This relation is of approximative nature and the constants \(a, b\) and \(c\) are generally calibrated by fitting measured \(^3\)-values for molecules whose dihedral angles are known from crystal structures\(^10-13\) or inferred from NMR data\(^14\). Since this Karplus relation is multi-valued for almost all except the very large and very small \(^3\)-values and the average \(\bar{J}(\theta(\mathbf{r}))\) is very nonlinear with respect to the average in \(\theta\), restraining using a standard penalty function may lead to unrealistic results\(^5,6,8\). Moreover, high-energy barriers between different conformations or \(\theta\)-angle values may inhibit a proper sampling of the various \(\theta\)-angle ranges that contribute to the measured averaged \(^3\)-values. These features of the relation between \(^3\)-values and dihedral angles have made their use in biomolecular structure refinement problematic. Here a solution to this problem is proposed.

In the next section, the new restraining method is explained, followed by an application of \(^3\)-value restraining to hen egg-white lysozyme and by a short discussion.
7.3 Theory

During a molecular dynamics simulation, the current (instantaneous) and average $^3J$-coupling constants can be monitored. For this, the $^3J$-values are expressed in terms of dihedral angles $\phi$ that are defined by non-hydrogen atoms of the molecule. Such an angle $\phi$ differs by a phase shift $\delta$ from the angle $\theta$ ($\theta = \phi + \delta^{15,16}$). The average is calculated using an exponentially decaying memory function, which results in a larger impact of recent $^3J$-values on the average,

$$
\overline{J}(t, \tau) = \frac{1}{\tau} \int_0^t \exp \left( -\frac{t' - t}{\tau} \right) J(t') dt'
$$

with $\tau$ the memory relaxation time (which determines how fast the memory decays) and $^3J(t)$ the calculated $^3J$-value at time $t^{3,7}$. If the average (Equation 7.4) and the experimental $^3J$-value do not match, a local (limited range) potential energy term for the dihedral angle corresponding to the particular $^3J$-coupling constant is introduced and increased in size until the dihedral angle changes value. In other words, as long as the calculated and experimental $^3J$-values do not match, the dihedral angle is forced away from the range of values that were sampled up till now in the simulation. This idea derives from local-elevation (le) search$^{17}$ in which the potential energy of already visited parts of configuration space is raised in order to avoid repetitive sampling of the same parts of configuration space in the simulation.

The mathematical and algorithmic formulation of the proposed method is the following. Whenever the simulated average of the $^3J$-value and the current $^3J$-value do not fulfill the experimental observation, the force constant of a penalty function, acting on the torsion angle $\phi(t)$ and its current value $\phi(t)$, is increased. The restraining potential energy function of a given ($k$-th) $^3J$-value is a sum of $N_{le}$ (local) terms

$$
V_{ki}^{res}(\phi_k(\mathbf{r}(t))) = \sum_{i=1}^{N_{le}} V_{ki}^{le}(\phi_k(t)),
$$

where, as in local-elevation conformational search$^{17}$, Gaussian functions centred at $\phi_k^0$ are used as (locally active, i.e. only around $\phi^0$) penalty terms:

$$
V_{ki}^{le}(\phi_k(\mathbf{r}(t))) = K^{res} w_{\phi_k}(t) \exp \left( - (\phi_k(t) - \phi_k^0)^2 / 2(\Delta \phi^0)^2 \right),
$$

where $w_{\phi_k}(t)$ is the weight of the $i$-th penalty function and $K^{res}$ the penalty function force constant. The centres $\phi_k^0$ of the Gaussian functions $V_{ki}^{le}$ are equally distributed over the range of possible values of $\phi_k$ ($\phi_k^0 = 2\pi i / N_{le}$ with $i = 1, ..., N_{le}$), and the width is given by $\Delta \phi^0 = 2\pi / N_{le}$.

The weight of the penalty function is accumulated during the simulation according to

$$
w_{\phi_k}(t) = t^{-1} \int_0^t \delta_{\phi_k(\mathbf{r}(t'))} \phi_k^0 \left( J(\phi_k(\mathbf{r}(t'))) - J_k^0 \right)^2 \left( \overline{J}(\phi_k(\mathbf{r}(t'))) - J_k^0 \right)^2 dt'.
$$
using a biquadratic term $^5 (J(\phi_k(t)) - J^0_k)^2 (\bar{J}(\phi_k(t)) - \bar{J}^0_k)^2$ to determine whether the $^3J$-value deviates from the experimentally observed one ($J^0_k$) with $\phi_k(r(t))$ being the torsion angle corresponding to the $^3J$-coupling constant $J_k$ and $\delta$ the Kronecker delta, which is defined using finite differences:

$$\delta_{\phi_k(t)\phi_k^0} = \begin{cases} 1 & \text{if } \phi_k^0 - \Delta\phi^0 / 2 \leq \phi_k(t) < \phi_k^0 + \Delta\phi^0 / 2 \\ 0 & \text{otherwise.} \end{cases}$$

Equation 7.7 ensures that the conformation is pushed away from $\phi_k^0$ unless either the average $\bar{J}(\phi_k(t))$ or the current value $J(\phi_k(t))$ are close to the experimental one, which leads sooner or later to an average close to the experimental $^3J$-value $j^0_k$.

It is straightforward to calculate the force resulting from $V^{Jres}_k$ on particle $q$:

$$f_q = -\frac{\partial}{\partial r_q}V^{Jres}_k = -\sum_{i=1}^{N_{le}}\frac{\partial}{\partial \phi_k}V^{le}_k \frac{\partial \phi_k}{\partial r_q} = \sum_{i=1}^{N_{le}}V^{le}_k \frac{\phi_k - \phi_k^0}{(\Delta\phi^0)^2} \frac{\partial \phi_k}{\partial r_q}.$$ 

Contrary to the original local-elevation method$^17$, using penalty functions to enforce restraints does not suffer from combinatorial explosion with increasing number of local-elevation degrees of freedom, as all restraints are treated independently.

In practice, flat bottom restraining can be achieved by only increasing the penalty function force constants $w_{\phi_k}(t)$ if the $^3J$-value deviates more than a given value $\Delta J^0$ from the experimental value $J^0_k$. For the instantaneous factor of the penalty function this leads to

$$V^{inst.Jres}(J(\phi_k(t))) = \begin{cases} (J(\phi_k(t)) - J^0_k - \Delta J^0)^2 & \text{for } J(\phi_k(t)) > J^0_k + \Delta J^0 \\ (J(\phi_k(t)) - J^0_k + \Delta J^0)^2 & \text{for } J(\phi_k(t)) < J^0_k - \Delta J^0 \\ 0 & \text{otherwise,} \end{cases}$$

and accordingly for the time-averaging factor

$$V^{avg.Jres}(\bar{J}(\phi_k(t))) = \begin{cases} (\bar{J}(\phi_k(t)) - J^0_k - \Delta J^0)^2 & \text{for } \bar{J}(\phi_k(t)) > J^0_k + \Delta J^0 \\ (\bar{J}(\phi_k(t)) - J^0_k + \Delta J^0)^2 & \text{for } \bar{J}(\phi_k(t)) < J^0_k - \Delta J^0 \\ 0 & \text{otherwise.} \end{cases}$$

### 7.4 Methods and results

The protein hen egg-white lysozyme was recently used to validate$^{18,19}$ the GROMOS$^{15,16,20}$ 53A6 force field$^{21}$, and the 45A3 parameter set$^{22}$. Using unrestrained MD simulations with explicit solvent, out of a hundred $^3J_{\alpha\beta}$-coupling constants, 31 showed a deviation from the experimental value$^{23}$ which was higher than 2 Hz, 11 a deviation higher than 3 Hz$^{19}$.

Here only a subset of 37 $^3J_{\alpha\beta}$-coupling constants that were assigned stereospecifically$^{23}$, which can therefore be used in $^3J$-value restraining, were considered (see Table 7.1).
Table 7.1: Subset of 37 $^3J_{\alpha\beta}$-coupling constants (in Hz) which could be assigned stereospecifically for hen-egg white lysozyme and are used in (local-elevation) $^3J$-value restraining. The $\beta_2$ and $\beta_3$ protons are defined according to standard rules. The experimental error is about 1 Hz.

We note that it is the $\chi_1$ side-chain torsional angle that plays the role of the restrained angle $\phi$ in Equations 7.5 to 7.11. Short simulations in vacuo using the X-ray structure as starting configuration showed 14 $^3J$-coupling constants with a deviation higher than 2 Hz. Trying to reduce the deviation from experiment using $^3J$-value restraining was only partially successful. We did not succeed in finding a good value of the force constant for any one of the restraining methods (instantaneous, time-averaged or biquadratic restraining penalty function) that would bring all $^3J$-coupling constants close to the experimental values without seriously changing the secondary structure. Two issues needed to be addressed: First, to be able to use a minimal restraining force constant, the latter should be adjusted individually for each $^3J$-value restraint. Second, to reproduce the experimental $^3J$-value and escape local minima of the physical or restraint po-
tential energy surface enhanced sampling of the corresponding torsional angle may be required. Adaptive restraints using local-elevation satisfy both requirements, as the force constant of the restraining penalty function is, if necessary, slowly built up during the simulation for each restraint. Furthermore, the restraining is achieved by pushing the simulation away from already visited conformations with $^3J$-coupling constants different from the experimental ones. In other words, sampling is enhanced for dihedral angle degrees of freedom with wrong $^3J$-values.

First, the sensitivity of the method with regard to the parameters $K^{\text{res}}$, $\Delta \phi^0$ and $N_{\text{le}}$ was investigated. From a short 100 ps unrestrained simulation of lysozyme in vacuo, using a time step size of 2 fs and constraining bond lengths by the SHAKE algorithm, an average violation of 1630 NOE distance upper bounds of 0.016 nm and a root-mean-square deviation (RMSD) for the 37 selected $^3J_{\phi\beta}$-coupling constants of 3.2 Hz were obtained. Then, a total of 36 simulations, each starting from the X-ray structure and lasting 100 ps, with all combinations of values for $K^{\text{res}} = 0.1, 0.05, 0.01, 0.005 \text{ Hz}^{-4}$, $\Delta \phi^0 = 0.5, 0.75, 1.0 \text{ Hz}$ and $\Delta \phi^0 = 30^0, 20^0, 10^0$ ($N_{\text{le}} = 12, 18, 36$), were used to determine whether significant improvement in the RMSD for the $^3J_{\phi\beta}$-coupling constants could be obtained without disrupting the structure, measured by the average violation of 1630 experimental NOE distance upper bounds (see Figure 7.1). Colours in the figure correspond to equal force constants (red for $K^{\text{res}} = 0.1 \text{ Hz}^{-4}$, blue for 0.05 $\text{ Hz}^{-4}$, yellow for 0.01 $\text{ Hz}^{-4}$ and green for 0.005 $\text{ Hz}^{-4}$), and line-styles to an equal number of intervals (solid for 12 intervals, dashed for 18 and dotted for 36) or local-elevation Gaussians per dihedral angle. The three values connected by a line use, from low to high $^3J$-value RMSD, an allowed deviation of $\Delta \phi^0 = 0.5 \text{ Hz}$ for the first, $\Delta \phi^0 = 0.75 \text{ Hz}$ for the second and $\Delta \phi^0 = 1.0 \text{ Hz}$ for the last value. Using 36 intervals, all simulations did better or equal in NOE violations (the value of the unrestrained simulation is indicated by the dotted black line), and even with an allowed deviation of $\Delta \phi^0$ of 1 Hz, satisfactory $^3J_{\phi\beta}$-coupling constants were obtained. Using sufficiently small force constants and enough intervals resulted in lower average NOE upper bound violations.

In Figure 7.2 the root-mean-square deviation over the set of 37 selected $^3J$-coupling constants during 100 ns of MD simulation of lysozyme in vacuo is shown. The solid black line, denoting an unrestrained simulation, shows an increase in deviation from experiment. All other lines correspond to adaptively restrained simulations, the thin line with a force constant $K^{\text{res}}$ of 0.005 $\text{ Hz}^{-4}$ and an acceptable deviation of $\Delta \phi^0 = 1.0 \text{ Hz}$, the dashed one with a force constant of 0.1 $\text{ Hz}^{-4}$ and an acceptable deviation of 0.5 Hz and the dotted one with $K^{\text{res}} = 0.05 \text{ Hz}^{-4}$ and $\Delta \phi^0 = 0.75 \text{ Hz}$. All use 36 intervals to discretize $\phi$ ($N_{\text{le}} = 36$). The time-averaging memory relaxation time $\tau$ (Equation 7.4) used in all restrained simulations was 5 ps. All combinations of parameters improve the RMSD of the $^3J$-coupling constants within the first 30 ps. The longer the simulation is, the lower the force constant $K^{\text{res}}$ may be to perturb the system as little as possible.

Comparing the evolution of selected angles $\phi$ during the simulation, three observations can be made: First, when starting from a configuration with a $^3J$-coupling constant far from the experimental value, rotation around the corresponding dihedral angle is immediate. An example
Figure 7.1: Average of the NOE distance upper bound violations as a function of the root-mean-square deviation (RMSD) of a set of 37 selected (see Table 7.1) $^3J_{\alpha\beta}$-coupling constants from experimental values$^{23}$ for different force constants $K_{\text{res}} = 0.100, 0.050, 0.010, 0.005 \text{Hz}^{-1}$ and different number of intervals $N_{\text{le}} = 12, 18, 36$ with corresponding $\Delta \phi^0 = 30, 20, 10$ degree. On each line the first value represents an allowed deviation of $\Delta \phi^0 = 0.50 \text{Hz}$, the second $\Delta \phi^0 = 0.75 \text{Hz}$ and the third $\Delta \phi^0 = 1.00 \text{Hz}$. The average NOE bound violation of the free simulation is indicated by the dotted black line. The RMSD of the $^3J_{\alpha\beta}$-coupling constant in the free simulation is $3.2 \text{Hz}$.
Figure 7.2: Root-mean-square deviation (RMSD) of a set of 37 selected (see Table 7.1) $^3J_{\alpha\beta}$-coupling constants from experimental values\textsuperscript{23}. The solid line is the root-mean-square deviation during an unrestrained simulation. All other lines are from simulations making use of local-elevation adaptive restraints. The thin line denotes use of a force constant $K_{J\text{res}} = 0.005\,\text{Hz}^{-4}$ and an allowed deviation of $\Delta J^0 = 1.0\,\text{Hz}$, the dashed line use of a force constant $K_{J\text{res}} = 0.1\,\text{Hz}^{-4}$ and an allowed deviation of $\Delta J^0 = 0.5\,\text{Hz}$, the dotted line use of a force constant $K_{J\text{res}} = 0.05\,\text{Hz}^{-4}$ and acceptable deviation of $\Delta J^0 = 0.75\,\text{Hz}$. All are using $N_{\text{int}} = 36$ intervals to discretise $\phi$, corresponding to a $\Delta \phi^0 = 10^\circ$.

$^3J$-value is close to the experimental one from the beginning, the corresponding dihedral angle is kept restrained to its value (panel A: $^3J_{\alpha\beta}$ of Thr(51), experimental: 9.3 Hz, unrestrained 4.3 Hz, restrained: 9.7 Hz). The root-mean-square fluctuation of the torsion angle $\phi$, once the correct conformation is found, is in the same order of magnitude for the restrained as for the unrestrained simulations. The time series and final (after 100 ps) restraining potential energy functions for all 37 $\chi_1$ torsional angles are shown in supplementary information.

Using restraints, the atom-positional root-mean-square deviation from the initial (X-ray) structure (considering only backbone atoms) decreases from 0.15 nm (unrestrained) to 0.12 nm in the restrained simulation (see Figure 7.4 and Figure 7.5). The atom-positional fluctuations are
Figure 7.3: The local-elevation potential energy functions for four selected $^3J_{\alpha\beta}$-coupling constants at the end of 100 ps of an adaptively restrained simulation ($K^{\text{res}} = 0.005 \text{ Hz}^{-4}$, $\Delta J^0 = 1.0 \text{ Hz}$ and $N_{t_e} = 36$) are shown (Thr(51), Thr(69), Thr(89) and Val(109) corresponding to panel A, B, C and D, upper half). The lower half depicts the time series (0.2 ps intervals) of the corresponding dihedral angle, black points indicate values obtained from the $^3J$-value restraining simulation, red points those from an unrestrained simulation. The experimental and average $^3J$-coupling constants are listed in Table 7.2.
Table 7.2: Experimental (exp) \(^3\)J\(_{\alpha\beta}\)-coupling constants (in Hz; error about 1 Hz) and values obtained from 100 ps of unrestrained (unrestr) and from 100 ps of restrained (restr) simulation using adaptive (local-elevation) \(^3\)J-value restraints with a set of 37 selected (see Table 7.1) \(^3\)J\(_{\alpha\beta}\)-coupling constants for four residues that show large deviation between the values obtained from the free simulation and the experimental ones.

<table>
<thead>
<tr>
<th>residue name</th>
<th>number</th>
<th>exp</th>
<th>unrestr</th>
<th>restr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thr</td>
<td>51</td>
<td>9.3</td>
<td>4.3</td>
<td>9.7</td>
</tr>
<tr>
<td>Thr</td>
<td>69</td>
<td>9.3</td>
<td>12.5</td>
<td>9.8</td>
</tr>
<tr>
<td>Thr</td>
<td>89</td>
<td>9.5</td>
<td>2.5</td>
<td>9.9</td>
</tr>
<tr>
<td>Val</td>
<td>109</td>
<td>8.0</td>
<td>3.2</td>
<td>8.2</td>
</tr>
</tbody>
</table>

Comparable. The secondary structure assignment shows no major loss in the overall structure of lysozyme, even though vacuum boundary conditions were used.

For the restrained simulation (\(K_{\text{res}} = 0.005 \text{Hz}^{-4}, \Delta J^0 = 1.0 \text{Hz}\) and \(N_{le} = 36\)), a total of 1630 inter-proton distances corresponding to NOE intensities\(^{19,27}\) have been analysed. We note that this set was the result of a slight revision\(^{19}\) of a set of 1632 NOE intensities\(^{27}\). These distance upper bounds include pseudo-atom corrections\(^{29}\) and the distances were determined from the simulations using \(r^{-3}\) averaging\(^{30}\). Their distribution is shown in Figure 7.6 as distance bound violations, i.e., distances averaged over the simulation minus the corresponding NMR derived upper distance bound. This difference can also adopt negative values, which means that in the MD simulations the inter-proton distance is on average shorter than the upper bound derived from the NMR experiment. The black bars show the distribution of the simulation using \(^3\)J\(_{\alpha\beta}\)-value restraining, red bars show the unrestrained distribution. Different from standard restraining simulations (using instantaneous or time-averaged \(^3\)J-value restraints) less NOE violations are observed when using the adaptive (local-elevation) restraining method for \(^3\)J\(_{\alpha\beta}\)-values presented here.

The dependence of the results on the initial structure of the simulation can be tested by repetitively using adaptive \(^3\)J-coupling constant restraining followed by an unrestrained simulation period. Figure 7.7 shows that no improvement of the root-mean-square deviation of the \(^3\)J-values during the unrestrained parts of the simulation is obtained. The simulations were carried out using a 20 ps and a 40 ps interval (with restraints switched on first, then switched off, etc.). The RMSD of \(^3\)J-values immediately increases when the restraints are switched off. This may have two causes. First, even a 40 ps restraining period may not be enough to relax the structure (in vacuo), or, second, the force field does not properly favour the experimental conformation of the 37 side-chain angles and needs the adaptive restraining penalty function to correct for this.
7.5 Discussion

A new application of the local-elevation simulation technique\textsuperscript{17} to achieve $^3J$-value restraining was presented. Using this method, it is possible to successfully restrain $^3J$-coupling constants without destabilising the overall molecular structure. In the example of lysozyme, even an improvement of reproducing experimental NOE distance bounds was observed. It can be applied to dihedral angles other than $\chi_1$, for which experimental $^3J$-values are available\textsuperscript{31}.

The method achieves selectively enhanced sampling by disfavouring conformations of dihedral angles with $^3J$-coupling constants deviating from experiment. Also, through the slow build up of the adaptive (local-elevation) potential energy penalty functions, a minimum of interference of the restraints compared to an unrestrained simulation is guaranteed. Furthermore, the method is not very sensitive with respect to the force constant and number of dihedral-angle intervals chosen, making it suitable to include $^3J$-value restraining in standard biomolecular NMR.

**Figure 7.4:** The time series of secondary structure elements during an unrestrained simulation of lysozyme is shown.
Figure 7.5: The time series of secondary structure elements during a (local-elevation) $^3J$-value restraining simulation using a set of 37 $^3J$-coupling constant restraints with adaptive force constants of lysozyme using a force constant $K^{\text{Jres}} = 0.005 \text{Hz}^{-4}$, $\Delta f^0 = 1.0 \text{Hz}$ as acceptable deviation and $N_{te} = 36$ is shown.

structure refinement.

7.6 Acknowledgements

Financial support by the National Center of Competence in Research (NCCR) Structural Biology of the Swiss National Science Foundation (SNSF) is gratefully acknowledged.
Figure 7.6: Violations of 1630 NOE distance upper bounds from a 100 ps MD simulation of lysozyme are shown. Black bars correspond to a simulation using (local-elevation) $^3J$-value restraining with 37 $^3J_{\alpha\beta}$-coupling constant restraints ($K^{\text{fres}} = 0.01 \text{Hz}^{-4}$ and $\Delta f^0 = 1.0 \text{Hz}$), white bars correspond to an unrestrained simulation.
Figure 7.7: The time series of the average root-mean-square deviation from the experimental $^{3}J_{\alpha \beta}$-values for a set of 37 $^{3}J_{\alpha \beta}$-coupling constants is shown. In the simulation corresponding to the solid line, adaptive $^{3}J$-coupling constant restraints ($K^{\text{res}} = 0.005 \text{Hz}^{-4}$, $\Delta J^{0} = 1.0 \text{Hz}$ and $N_{\text{fe}} = 36$) were switched off for the first 20 ps, then on for 20 ps, off again for the next 20 ps and so on. In the simulation denoted by the dotted black line, 40 ps intervals were used.
7.7 Bibliography


196 Chapter 7. Adaptive restraints using local-elevation simulation


Chapter 8

Approximate flexible distance constraints

8.1 Summary

A fast but approximative method to apply flexible constraints to bond lengths in molecular dynamics simulations is presented and the effects of the approximation are investigated. The method is not energy conserving, but coupling to a temperature bath results in stable simulations. The high frequencies from bond-length vibrations are successfully removed from the system while maintaining the flexibility of the bonds. As a test liquid neopentane is simulated at different pressures. Energetic and dynamic properties are not affected by the new flexible constraint simulation method.
Chapter 8. Approximate flexible distance constraints

8.2 Introduction

Nowadays large molecular systems can be studied theoretically at atomic detail using molecular dynamics (MD) simulations or the Monte Carlo method. In classical MD simulations the motion of the particles is governed by Newton's equations of motion. The time step size in MD simulations is limited by the fastest vibrations in the system, normally bond-length vibrations. Therefore, it is not surprising that there are quite a few methods available to constrain bonds to their ideal lengths and thus removing these high frequencies. Constraining bonds works rather well, sometimes even for thermodynamic states quite different from the one used in fitting the bond-length parameters. However, processes and circumstances have been identified where the flexibility of bonds can be crucial. This might be especially true upon variation of the pressure or in close confinements, as for water molecules in an ion-channel. For such cases water models that include flexibility have been proposed. However, flexible models add very fast vibrations to the system thereby again limiting the time step. In addition, these vibrations are only loosely coupled to the other degrees of freedom making long equilibration times necessary.

To overcome these shortcomings flexible constraint algorithms have been proposed. Using these methods the bond-length distance constraints are (adiabatically) adjusted to their current minimum-energy lengths according to the total energy (or total potential energy, including the bond-stretching energy) of the system at the current time (step). The methods proposed so far require multiple energy evaluations at every time step and thus are an order of magnitude more time-consuming than hard-constraint algorithms. In this article we propose an approximation that leads to a fast flexible constraint algorithm which is as computationally efficient as the SHAKE method to impose hard constraints, and evaluate its effects on the physical properties of the simulated systems.

In the following sections we will first briefly describe hard constraints and then introduce the flexible constraint method. Afterwards we apply the method to a very simple test system and then present the results from simulations of liquid neopentane.

8.3 Hard constraints

Classical MD simulations are governed by a Hamiltonian of type

\[ H(q, p) = \frac{p^T M^{-1} p}{2} + U(q) \]  

with \( q \) and \( p \) the positions and momenta of the particles, \( M \) the symmetric mass matrix and \( U \) a potential energy function.

When applying constraints to the system using the Lagrange multiplier technique, it is convenient to split the potential energy function \( U \) into a part \( U_{nc} \) describing the unconstrained
8.4. Flexible constraints

interactions (dihedral torsion, Lennard-Jones and electrostatic interactions, unconstrained bond angles, etc.) and a part $U_c$ representing the constraint forces,

$$U(q) = U_{nc}(q) + U_c(q)$$

$$U_c(q) = \sum_{k=1}^{N_c} \lambda_k g_k(q)$$

with

$$g_k(q) \equiv |r_{k1} - r_{k2}|^2$$

and the $N_c$ Lagrange multipliers $\lambda_k$ determined by the $N_c$ constraint conditions $g_k(q) = 0$. $r_k$ is the vector connecting the atom pair $k_1$ and $k_2$ of constraint $k$ ($r_k = r_{k1} - r_{k2}$) and $d_k^{(0)}$ the (ideal) constraint length.

It is possible to solve for the constraint forces $f_i = -\sum_{k=1}^{N_c} \lambda_k g_k(q)$ using the unconstrained forces and positions resulting from these forces. A widely used method to this end is the SHAKE algorithm. It efficiently accomplishes this by decoupling the constraints and linearising the quadratic equation to solve for the Lagrange multipliers. This procedure is then repeated until all the constraints are satisfied to the required accuracy. For each iteration, the Lagrange multipliers are given by

$$\lambda_k = \frac{(d_k^{(0)})^2 - (r_{k}^{nc}(t + \Delta t))^2}{-4(\Delta t)^2 (m_{k1}^{-1} + m_{k2}^{-1}) (r_k(t) \cdot r_k^{nc}(t + \Delta t))}$$

with the non-constrained or free flight positions at time $t + \Delta t$ defined via the leap-frog scheme through $q^{nc}(t + \Delta t) = q(t) + m^{-1} (p(t - \Delta t/2) \Delta t + f^{nc}(t)(\Delta t)^2)$ and the non-constrained or free forces at time $t$ through $f^{nc}(t) = -\nabla_q U_{nc}(q(t))$.

8.4 Flexible constraints

A simple way of implementing flexible constraints is by recalculating the ideal constrained bond lengths for every constraint at each time step. These new constraint lengths are calculated using the forces resulting from the interactions of the constrained pair of atoms with all other particles. In other words the sum of the total energy ($U_{nc}$) and a hypothetic harmonic bond-stretching energy $U^{constr} = \sum_{k=1}^{N_c} U_k^{constr}$ is minimised with respect to the bond or constraint lengths $d_k$ at each time step. This means that the hypothetical forces from the bond-stretching terms have to exactly compensate the forces from the real potential energy terms ($U_{nc}$) acting along the constraint directions,

$$K_k \left( d_k - d_k^{(0)} \right) = F_k$$

(8.6)
with $F_k$ the force along constraint $k$, $K_k$ and $d_k^{(0)}$ the force constant and the zero energy distance of the flexible constraint of the form $U^\text{constr}_k(q) = \frac{1}{2}K_k(|r_k| - d_k)^2$ and $d_k$ the adaptable constraint length.

The force acting on each constraint $k$ is estimated from the hypothetical change in the constraint length (during unconstrained or free flight) due to the interaction $U_{nc}$,

$$F_k = \frac{\mu_k}{\Delta t^2} \cdot \left( |r_k - (\Delta t)^2 m_k^{-1} \nabla_{q_k} U_{nc}(q)| + \right)
(\Delta t)^2 m_k^{-1} \nabla_{q_k} U_{nc}(q) - |r_k| \right) \quad (8.7)$$

with $\mu_k = \frac{m_k - m_k^2}{m_{i_1} + m_{i_2}}$ the reduced mass of the constraint $k$, $\Delta t$ the time step of the leap-frog discretisation, $m_i$ the mass of atom $i$, and $-\nabla_{q_i} U_{nc}$ the force on atom $i$ due to the unconstrained part of the potential energy $U$.

It is possible to also include the change in constraint length due to the kinetic energy term $\frac{1}{2}p^T M^{-1} p$. To obtain Equation 8.7, the potential energy was split into an unconstrained and a constrained part and only the former part was used. It is possible to also split the kinetic energy term in an unconstrained and a constrained part, and then to save the size of the velocity, $v_k$, of the constraint length change from the previous step. The change in constraint length due to this (initial) velocity is then subtracted from the total change of the constraint length, leading to a force on constraint $k$ given by

$$F'_k = \frac{\mu_k}{\Delta t^2} \cdot \left( |r_k + \Delta t m_k^{-1} p_k - (\Delta t)^2 m_k^{-1} \nabla_{q_k} U_{nc}(q)| - \right)
\Delta t m_k^{-1} p_k + (\Delta t)^2 m_k^{-1} \nabla_{q_k} U_{nc}(q) - v_k \Delta t - |r_k| \right) \quad (8.8)$$

with $p_i$ the momentum of atom $i$ and $v_k \Delta t$ the change of the length of the flexible constraint $k$ from the previous (leap-frog) step. Below we formulate the flexible constraint algorithm using $F_k$ for the external force, but this can also be done with $F'_k$. In the tests the algorithm with $F'_k$ has been used.

It is now easy to calculate from Equation 8.6 the constraint lengths for which the energy becomes minimal, i.e. for which the hypothetical harmonic constraint forces oppose and compensate the external forces $F_k$,

$$d_k = \frac{F_k}{K_k} + d_k^{(0)}. \quad (8.9)$$

This results in the flexible constrained Hamiltonian system
8.4. Flexible constraints

\[
\frac{d}{dt} q = M^{-1} p \tag{8.10}
\]

\[
\frac{d}{dt} p = -\nabla_q U_{nc}(q) - \nabla_q \sum_{k=1}^{N_c} \lambda_k(t) g'_k(q,t) \tag{8.11}
\]

\[
o = g'_k(q,t) \quad k = 1, 2, \ldots, N_c \tag{8.12}
\]

with \( g'_k(q,t) = |r_k|^2 - d_k^2(t) \). These modified equations of motion can still be discretised using the SHAKE method. We note that \( g'_k(q,t) \) through \( d_k(t) \) is dependent on the gradient of the non-constrained potential energy \( U_{nc} \). Therefore, the computation of the Hessian of \( U_{nc} \) is required. As this only involves the non-constrained part of the potential energy, no recalculation of the constraint lengths during the SHAKE iterations is necessary.

The constraint forces are given by

\[
f'_i(t) = -\nabla_q \sum_{k=1}^{N_c} \lambda_k g'_k(q,t)
\]

\[
= -2 \sum_{k=1}^{N_c} \lambda_k [r_k(t)(\delta_{ik1} - \delta_{ik2}) - d_k(t + \Delta t)\nabla_q d_k(t + \Delta t)] \tag{8.13}
\]

Using Equation 8.13 the new constrained positions are

\[
r_i(t + \Delta t) = r_i^{nc}(t + \Delta t) - \frac{2(\Delta t)^2}{m_i} \sum_{k=1}^{N_c} \lambda_k [r_k(t)(\delta_{ik1} - \delta_{ik2}) - d_k(t + \Delta t)\nabla_q d_k(t + \Delta t)] \tag{8.14}
\]

Inserting Equation 8.14 into the constraint conditions Equation 8.12 yields (neglecting terms of second order in \( \lambda_k \))

\[
o = (r_k(t + \Delta t))^2 - (d_k(t + \Delta t))^2
\]

\[
= (r_k^{nc}(t + \Delta t))^2 - 4(\Delta t)^2 \lambda_k r_k^{nc}(t + \Delta t) \cdot \left[ \left( m_{k1}^{-1} + m_{k2}^{-1} \right) r_k(t) - d_k(t + \Delta t) \left( m_{k1}^{-1}\nabla q_{k1} d_k(t + \Delta t) - m_{k2}^{-1}\nabla q_{k2} d_k(t + \Delta t) \right) \right]
\]

\[
- (d_k(t + \Delta t))^2 + O(\lambda_k^2). \tag{8.15}
\]

Solving for \( \lambda_k \) results in

\[
\lambda_k = \left( (d_k(t + \Delta t))^2 - (r_k^{nc}(t + \Delta t))^2 \right)
\]

\[
- 4(\Delta t)^2 r_k^{nc}(t + \Delta t) \cdot \left[ \left( m_{k1}^{-1} + m_{k2}^{-1} \right) r_k(t) - d_k(t + \Delta t) \right]
\]

\[
\left( m_{k1}^{-1}\nabla q_{k1} d_k(t + \Delta t) - m_{k2}^{-1}\nabla q_{k2} d_k(t + \Delta t) \right) \right]^{-1} \tag{8.16}
\]

This system of \( N_c \) equations can be solved iteratively using the SHAKE method. It has to be noted that each constraint update does not only involve the two atoms forming the constraint but also
all other atoms interacting with these. Because generally all pairs of atoms in the system interact, all constraints become coupled in this scheme.

Below we investigate the effects of approximating the constraint force due to constraint \( k \) on particle \( i \) (Equation 8.13) by

\[
\mathbf{f}_i^c(t) = -2\lambda_k \left( \delta_{i,k_1} - \delta_{i,k_2} \right) \mathbf{r}_k(t)
\]  

(8.17)

neglecting the change of \( d_k(t) \). This approximation leads to weakly coupled constraint equations and allows the use of the SHAKE algorithm without any further calculation than adapting the constraint length \( d_k(t) \) to the current external forces using Equation 8.9.

A closer look at the neglected term yields

\[
-\nabla q_i \lambda_k \left( - (d_k(t + \Delta t))^2 \right) = 2\lambda_k d_k(t + \Delta t) \nabla q_i d_k(t + \Delta t)
\]  

(8.18)

with (using Equations 8.9 and 8.7)

\[
\nabla q_i d_k(t + \Delta t) = \frac{\mu_k}{K_k(\Delta t)^2} \left[ \frac{\mathbf{r}_k^{nc}(t + \Delta t)}{|\mathbf{r}_k^{nc}(t + \Delta t)|} \cdot \left( \nabla q_i \otimes (\mathbf{r}_k(t) - m_{k_1}^{-1}(\Delta t)^2 \nabla q_{k_1} U_{nc}(\mathbf{q}(t)) + m_{k_2}^{-1}(\Delta t)^2 \nabla q_{k_2} U_{nc}(\mathbf{q}(t))) \right) \right.
\]  

(8.19)

and

\[
\mathbf{r}_k^{nc}(t + \Delta t) = \mathbf{r}_{k_1}(t) + \mathbf{p}_{k_1}(t + \Delta t/2) m_{k_1}^{-1}(\Delta t) - m_{k_1}^{-1}(\Delta t)^2 \nabla q_{k_1} U_{nc}(\mathbf{q}(t))
\]  

\[ - \mathbf{r}_{k_2}(t) - \mathbf{p}_{k_2}(t + \Delta t/2) m_{k_2}^{-1}(\Delta t) + m_{k_2}^{-1}(\Delta t)^2 \nabla q_{k_2} U_{nc}(\mathbf{q}(t)),
\]  

(8.20)

where the \( \otimes \) operator denotes the tensor product. To evaluate Equation 8.19 one needs second derivatives of the potential energy function \( U_{nc}(\mathbf{q}) \). Equations 8.18 and 8.19 show that a larger force constant \( K_k \) or a smaller reduced mass \( \mu_k \) lead to a smaller error of the approximation. Collecting the gradient terms and the position dependent terms in Equation 8.19 yields

\[
\nabla q_i d_k(t + \Delta t) = \frac{\mu_k}{K_k(\Delta t)^2} \left[ \frac{\mathbf{r}_k^{nc}(t + \Delta t)}{|\mathbf{r}_k^{nc}(t + \Delta t)|} \cdot \left( -m_{k_1}^{-1}(\Delta t)^2 \nabla q_{k_1} U_{nc}(\mathbf{q}(t)) + m_{k_2}^{-1}(\Delta t)^2 \nabla q_{k_2} U_{nc}(\mathbf{q}(t)) \right) \right]
\]  

(8.21)

This shows that the error in the forces acting on all the atoms interacting with the atoms \( k_1 \) and \( k_2 \) forming constraint \( k \) is independent of the size of the time step. If the change in the direction
of the constraint \( \frac{\Delta r_{ik}}{\Delta t} - \frac{F_k}{m_i} \) is proportional to \((\Delta t)^2\), also the last term in Equation 8.21, which involves only the constrained atoms, is time-step independent (which is expected for forces).

The flexible constraint algorithm Equations 8.7 to 8.17 will be stable if the changes in the constraint lengths \(d_k\) per time step are smaller than the movement of the atoms induced by the non-constrained forces. The algorithm may suffer from instability if the constraint forces happen to be large and the time step is very small. The large constraint forces \(F_k\) will induce, through Equation 8.9, large adjustments of the flexible constraint lengths \(d_k\) (regardless of the time step), while the small time step will induce (much) smaller atomic movements due to the non-constrained forces. This may happen when a flexible constraint simulation is initialised by starting from a hard constrained one with large constraint forces.

Using flexible constraints a slight complication arises with respect to the calculation of the temperature from the kinetic energy in the non-constrained degrees of freedom. As flexible constraints change with time, there will be (small) velocities along the constraint directions. But in analogy to hard constraints the flexible constraints are not counted as degrees of freedom. Therefore, to correctly calculate the temperature of the non-constrained degrees of freedom the kinetic energy along the flexible constraints is subtracted from the total kinetic energy before calculating the temperature.

## 8.5 Numerical experiments

Approximative flexible constraints were implemented in the Groningen Molecular Simulation package (GROMOS)\textsuperscript{21,22}. In all simulations the constraints were imposed using the SHAKE method with a geometric tolerance of \(10^{-6}\). To prove the correctness of the implementation free rotors have been simulated. As there is no physical interaction \(U_{nc}\) the first term in Equation 8.21 is zero and the size of the second term can be controlled by the speed of the rotation. Table 8.1 shows the results for different rotors and time steps. It can be seen that the simulation is stable with respect to changes of the time step used in the leap-frog integration if hard or flexible constraints are applied. The flexible constraint simulation at 0.5 fs illustrates the stability problems occurring due to the start from a hard constrained position. Furthermore, it is clear that increasing the force constant of the bond drives the flexible constrained solution towards the hard constrained solution. Changes of the mass of the atoms are a bit more difficult to interpret as the initial kinetic energy of the system is different, because no changes to the initial velocities were made. Therefore, the system is simulated at a different temperature. Not surprisingly, the energy loss is higher for higher temperature simulations. Not unexpectedly, halving the mass has the same effect on the average bond length as doubling the harmonic force constant, both in the unconstrained and in the flexible constrained case.
<table>
<thead>
<tr>
<th>method</th>
<th>Δt</th>
<th>K</th>
<th>mass</th>
<th>bond length</th>
<th>total energy</th>
<th>energy loss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[ps]</td>
<td>[10^5 kJ/mol⁻¹ nm⁻²]</td>
<td>[amu]</td>
<td>[nm]</td>
<td>[kJ/mol]</td>
<td>%</td>
</tr>
<tr>
<td>unconstrained</td>
<td>0.002</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153098</td>
<td>2.5059</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>0.002</td>
<td>3.35</td>
<td>7.518</td>
<td>0.153049</td>
<td>1.2529</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>0.002</td>
<td>3.35</td>
<td>30.070</td>
<td>0.153195</td>
<td>5.0117</td>
<td>0.020</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>6.69</td>
<td>15.035</td>
<td>0.153049</td>
<td>2.5059</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>13.39</td>
<td>15.035</td>
<td>0.153024</td>
<td>2.5059</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>0.008</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153097</td>
<td>2.5059</td>
<td>0.036</td>
</tr>
<tr>
<td>hard constrained</td>
<td>0.002</td>
<td>-</td>
<td>15.035</td>
<td>0.153000</td>
<td>2.5059</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>0.032</td>
<td>-</td>
<td>15.035</td>
<td>0.153000</td>
<td>2.5059</td>
<td>0.000</td>
</tr>
<tr>
<td>flexible constrained</td>
<td>0.0005</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153098</td>
<td>2.5059</td>
<td>1.704</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153098</td>
<td>2.5059</td>
<td>0.128</td>
</tr>
<tr>
<td></td>
<td>0.002</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153098</td>
<td>2.5059</td>
<td>0.128</td>
</tr>
<tr>
<td></td>
<td>0.002</td>
<td>3.35</td>
<td>7.518</td>
<td>0.153049</td>
<td>1.2529</td>
<td>0.064</td>
</tr>
<tr>
<td></td>
<td>0.002</td>
<td>3.35</td>
<td>30.070</td>
<td>0.153194</td>
<td>5.0117</td>
<td>0.253</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153098</td>
<td>2.5059</td>
<td>0.128</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>3.35</td>
<td>7.518</td>
<td>0.153049</td>
<td>1.2529</td>
<td>0.064</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>3.35</td>
<td>30.070</td>
<td>0.153194</td>
<td>5.0117</td>
<td>0.253</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>6.69</td>
<td>15.035</td>
<td>0.153049</td>
<td>2.5059</td>
<td>0.064</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>13.39</td>
<td>15.035</td>
<td>0.153049</td>
<td>2.5059</td>
<td>0.032</td>
</tr>
<tr>
<td></td>
<td>0.008</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153098</td>
<td>2.5059</td>
<td>0.132</td>
</tr>
<tr>
<td></td>
<td>0.008</td>
<td>6.69</td>
<td>15.035</td>
<td>0.153049</td>
<td>2.5059</td>
<td>0.068</td>
</tr>
<tr>
<td></td>
<td>0.008</td>
<td>13.39</td>
<td>15.035</td>
<td>0.153025</td>
<td>2.5059</td>
<td>0.036</td>
</tr>
<tr>
<td></td>
<td>0.016</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153096</td>
<td>2.5059</td>
<td>0.128</td>
</tr>
<tr>
<td></td>
<td>0.032</td>
<td>3.35</td>
<td>15.035</td>
<td>0.153097</td>
<td>2.5059</td>
<td>0.128</td>
</tr>
</tbody>
</table>

Table 8.1: Free rotor. Average bond length, average total energy and energy loss as function of time step Δt, force constant K of the harmonic bond interaction, and masses of the two atoms. All simulations were in a (dynamic, for the unconstrained simulation) equilibrium state after 1 ps and covered at least 1 ps. The constraints were imposed using the SHAKE method with a geometric tolerance of 10⁻⁶. Total energy at start was 2.5059 kJ/mol for an atom mass of 15.035 amu, 1.2529 kJ/mol for 7.5175 amu and 5.0117 kJ/mol for 30.070 amu.
8.5. Numerical experiments

8.5.1 Ethane collision in the gas phase

A simulation of a collision between two ethane molecules was performed. At the start one ethane is aligned parallel to the y-axis with one of its atoms at the origin. The distance between the united CH\textsubscript{3} atoms is equal to the ideal bond length and the atoms are at rest. The second molecule lies on the x-axis at a distance of 1.2\,nm with its bond at the ideal value. It moves toward the first ethane at a speed of 2.0\,nm/ps. All atom masses and the two bond force constants and bond lengths are equal. The two ethanes interact according to a Lennard-Jones interaction using the 45A3 parameter set of the GROMOS force field\textsuperscript{23}. The system was simulated using unconstrained bonds and using flexible bond constraints. From Figure 8.1 one can see that the flexible constrained simulation reproduces the average bond lengths of the unconstrained simulation without the fast bond oscillations observed in the latter. After the collision the ethanes start spinning. In both simulations one can see the increased (average) bond length due to the internal (rotational) kinetic energy of the molecules.

![Figure 8.1: Ethane bond lengths as a function of time during a collision of two ethane molecules. Atomic masses are 15\,amu, the ideal bond lengths are 0.153\,nm and the harmonic bond force constant is 3.35\,10^5\,kJ\,mol^{-1}\,nm^{-2}. Initially ethane 1 is at rest and ethane 2 moves with a velocity of 2\,nm\,ps^{-1} towards it.](image)
8.5.2 Neopentane liquid

Molecular dynamics simulations were carried out for liquid neopentane. The system consists of 512 molecules in a cubic periodic box. At 1 atm neopentane has a melting point of 256.75 K and a boiling point of 282.63 K. The density at 298 K and increased pressure is 0.5852 g/ml. The simulations were performed at 273 K and a pressure of 1 kbar as well as at a pressure of 5000 kbar. Weak coupling to temperature and pressure baths\textsuperscript{24} with a coupling constant of 0.5 ps for pressure coupling and 0.1 ps for temperature coupling was applied. The pressure was calculated through a molecular virial unless specified otherwise.

First, the energy conservation as a function of the time-step size has been investigated (Figure 8.2). The drift in total energy for the simulations using flexible constraints results in larger fluctuations. Nevertheless the algorithm is fairly stable towards an increase in time-step size (even up to 19 fs for these short simulations of 20 ps), whereas hard constraints result in a SHAKE error at a time-step size of 15 fs. For the unconstrained simulation a time-step size of 1 fs is just acceptable, one should rather use only 0.5 fs even for a system consisting of only quite heavy atoms. Results from longer constant temperature simulations using flexible constraints are also shown. Up to a time-step size of 2 fs the fluctuations in total and kinetic energy are comparable to those for the hard constrained simulations. Larger time-steps lead through the coupling with the temperature bath to larger fluctuations of the total energy, although these are reduced for larger time-step sizes (6 to 8 fs). Systems with more variety in the degrees of freedom might suffer less from this side effect of the use of flexible constraints.

For the following liquid neopentane simulations a time-step size of 1 fs was used which is clearly fine for flexible constrained and hard constrained simulations. Also the unconstrained simulation trajectories obtained proved to be stable.

The non-bonded interactions were treated with a triple range cutoff scheme with a short cutoff of 0.8 nm and a long cutoff of 1.4 nm. The pairlist was updated every five time steps. All parameter values were taken from the 45A3 GROMOS force field.

Table 8.2 gives an overview of the energetic properties of the different simulations. From Table 8.3 one can see the effect the flexible constraints have on the internal kinetic energy. For this set of degrees of freedom the temperature is about 5 to 10 K lower than the average temperature of all degrees of freedom.

The thermostat tries to compensate the continuous loss and increases in this process the temperature of the translational (com) degrees of freedom by 10 to 14 K.

The density (Figure 8.3) converges within the first 20 ps and is stable for the rest of the simulation. When applying flexible constraints the translational kinetic energy of the neopentane molecules is on average higher than in an unconstrained (or hard constrained) simulation, because internal motion is suppressed. This leads to a higher pressure and, as the pressure is weakly coupled to a reference value, in the end to a lower density. To further investigate the sensitivity of the pressure calculation in regard to the treatment of the bonds, simulations at 1 and at
Table 8.2: Average energies and energy fluctuations of liquid neopentane obtained from 200 ps MD simulations at 273 K and different pressures. Constraints were imposed with a geometric tolerance of $10^{-6}$. The error estimate is calculated according to Allen and Tildesley.

Table 8.3: Average, root-mean-square fluctuation, and error estimate of the total temperature, the centre of mass translational temperature, and the internal and rotational temperature of the neopentane molecules at different pressures from 200 ps of MD simulation.
Figure 8.2: Total energy and kinetic energy root-mean-square fluctuations from 20 ps of microcanonical MD simulations at 273 K temperature and 1 kbar pressure as a function of the time-step size. Total energy fluctuations are denoted for unconstrained simulations by a solid line, for hard constrained simulations by a dash-dotted line, for flexible constraints by a thick dashed line. Kinetic energy fluctuations are denoted for unconstrained simulations by a dashed line, for hard constrained simulations by a dash-double-dotted line, for flexible constrained simulations by a dotted line. For comparison the results of 50 ps of constant-temperature (273 K) MD simulation at 1 kbar pressure with flexible constraints using different time-step sizes are indicated by circles (total energy fluctuations) and diamonds (kinetic energy fluctuations).

5000 kbar using either harmonic bonds, hard constraints or flexible constraints with a molecular virial or an atomic virial have been carried out. One can easily see that the average pressure does not depend on whether a molecular virial or an atomic virial is used. Due to possibly quite large forces during the initial period of a flexible constraints run (and back-coupling through box size changes due to weak pressure coupling), the flexible constraint simulations had to be started from pre-equilibrated structures (using first 1 ps of hard constrained, constant pressure and then 1 ps of flexible constrained, constant volume simulation). It seems that for the flexible constraint runs the pressure is a bit higher and the density therefore a bit lower when using an atomic virial
than when using a molecular one.

![Figure 8.3](image_url)

Figure 8.3: Density of liquid neopentane at \( T = 273 \text{ K} \) and different pressures. Panels a and b show simulations at a pressure of 1 kbar, panels c and d at a pressure of \( p = 5000 \text{ kbar} \). The panels on the left (a and c) show data obtained from simulations using a molecular virial, for the ones on the right (b and d) the data has been obtained using an atomic virial. A solid line means full flexible bonds, a dotted line stands for hard constraints and a dashed line for flexible constraints. The starting structures of the constrained simulations have already been equilibrated.

There is not much change in the average bond length due to the increased pressure (Table 8.4). For the unconstrained simulation the average bond length decreases by 0.14 %, for flexible constraints the decrease is 0.07 %. The difference might arise from a slightly changed distribution of the kinetic energies over the degrees of freedom (favouring the translational over the internal / rotational ones), from the different density and from the approximation made in calculating the flexible constrained bond lengths. To further investigate this deviation, the total forces on the atoms (not including the bond terms) were projected on the bonds for the different simulation algorithms (see Figure 8.4). One can see that the force acting on the bonds is much higher in unconstrained simulations than when applying hard constraints. As flexible constraints adapt to the environment, these forces get even smaller for them. This means that the force constant of the bond interaction term used to determine the flexible bond lengths has to be lowered to obtain
similar results as in unconstrained simulations for the (average) bond lengths. To demonstrate this a short (50 ps) flexible constrained MD simulation at 273 K and 5000 kbar pressure using half the force constant for the bond interaction term was carried out. The bond lengths shown in Table 8.4 and Figure 8.5 are in good agreement with those of the corresponding unconstrained MD simulation (with a bond length decrease of 0.15 % for the flexible constrained simulation) and display smaller fluctuations.

<table>
<thead>
<tr>
<th></th>
<th>unconstrained</th>
<th>flexible constrained</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 kbar</td>
<td>5000 kbar</td>
</tr>
<tr>
<td>bond length [nm]</td>
<td>0.153000</td>
<td>0.152781</td>
</tr>
<tr>
<td>rms fluctuation $10^{-3}$ [nm]</td>
<td>2.460</td>
<td>2.673</td>
</tr>
<tr>
<td>bond energy [kJ/mol]</td>
<td>2294</td>
<td>2521</td>
</tr>
</tbody>
</table>

Table 8.4: Average bond lengths and bond energies for liquid neopentane obtained at 273 K and different pressures from 200 ps of MD simulation. For comparison also the values obtained from a 50 ps flexible constrained MD simulation at 273 K temperature and 5000 kbar pressure with half the force constant for the bond potential energy term used to determine the flexible constrained bond lengths are shown in the right most column.

The average bond length does not change much for the low pressure simulation when using flexible constraints with an atomic virial. This again shows that the difference in density does not have a significant effect on the bond lengths.

In Figure 8.6 the bond-stretching and bond-angle bending vibrational spectra of liquid neopentane are displayed. The higher-frequency vibrations of the bond lengths vanish when applying flexible constraints. The change of the bond lengths follows adiabatically the unconstrained forces and therefore the lower bond-stretching frequency (at 11 ps$^{-1}$) corresponds to the bond-angle bending frequency. The modulation of the bond-angle bending vibration (at 41 ps$^{-1}$) by the bond-length change also disappears when applying flexible constraints. As in this simple system now all internal molecular degrees of freedom have the same frequencies, coupling between these is to be expected. Especially for the high pressure simulation this can be confirmed from the spectrum.

From the bond-length distributions (Figure 8.7) one can see a slight shift of the maximum towards shorter bond lengths at increased pressure. The width of the distribution increases with pressure. Both phenomena are reproduced when using flexible constraints. When applying flexible constraints the width of the distributions is reduced compared to unconstrained simulation, as expected.

A look at the radial distribution functions (Figure 8.8) shows that the changes in the bond-
8.6 Discussion

A fast but approximate algorithm to impose flexible constraints on bonds or atom-atom distances has been proposed. The method was shown to remove the fast frequencies of the bond vibrations from the system while still letting the bond lengths adapt to the environment. It could also qualitatively reproduce the changes in the bond-length distribution when higher pressure was applied.

**Figure 8.4:** Average of the projection of the total force (excluding the bond term) on the bonds during 1 ps of simulation of liquid neopentane at 273 K temperature and 5000 kbar pressure. The solid line denotes an unconstrained, the dotted line a hard constrained and the dashed line a flexible constrained MD simulation.

Length distributions do not have any visible influence on the radial distributions. Using the flexible constraint algorithm the diffusion coefficients (Table 8.5) are in the same range as the diffusion coefficients calculated from the hard constrained and the unconstrained simulations. The value obtained for hard constraints at high pressure is not completely converged.

8.6 Discussion
Figure 8.5: Bond-length distribution in liquid neopentane from 50 ps MD simulation at 273 K and 5000 kbar pressure. The solid line denotes an unconstrained, the dotted line a flexible constrained MD simulation.

\[
\begin{array}{lcl}
p = 1 \text{kbar} & p = 5000 \text{kbar} \\
D & D \\
10^{-6}[\text{cm}^2/\text{s}] & 10^{-6}[\text{cm}^2/\text{s}]
\end{array}
\]

<table>
<thead>
<tr>
<th></th>
<th>( p = 1 \text{kbar} )</th>
<th>( p = 5000 \text{kbar} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>unconstrained</td>
<td>40.8 ( 10^{-6} \text{cm}^2/\text{s} )</td>
<td>1.6 ( 10^{-6} \text{cm}^2/\text{s} )</td>
</tr>
<tr>
<td>hard constrained</td>
<td>44.3 ( 10^{-6} \text{cm}^2/\text{s} )</td>
<td>0.9 ( 10^{-6} \text{cm}^2/\text{s} )</td>
</tr>
<tr>
<td>flexible constrained</td>
<td>37.7 ( 10^{-6} \text{cm}^2/\text{s} )</td>
<td>1.5 ( 10^{-6} \text{cm}^2/\text{s} )</td>
</tr>
</tbody>
</table>

Table 8.5: Diffusion coefficient of liquid neopentane at 273 K and different pressures. The diffusion coefficient has been calculated using the mean-square displacement of the molecules over 300 ps.
Figure 8.6: Bond-stretching vibrational spectrum of liquid neopentane from 200 ps MD simulation at 273K at low (1 kbar) pressure (a) and at high (5000 kbar) pressure (b). Bond-angle bending vibrational spectrum at low pressure (c) and at high pressure (d). Solid lines denote an unconstrained, dashed lines a flexible constrained simulation.

applied. Due to the approximations made to obtain a fast algorithm, the algorithm reproduces only 50% of the expected change of the average bond length (as compared to the non-constrained simulation). This can be compensated by the use of a lower force constant in the bond-energy term acting along the flexible constraint. The energy loss caused by the approximation Equation 8.17 can be counteracted using weak coupling to a temperature bath with standard coupling parameter values. However, because of the energy flow, equipartition of the kinetic energy among the degrees of freedom is not exactly fulfilled anymore. This affects the value of the pressure and, if pressure coupling is activated, the density.

Previously, alternative algorithms to impose flexible constraints have been proposed\textsuperscript{19,20}, which were computationally an order of magnitude more expensive than the standard hard constraint or unconstrained algorithms and therefore less suitable for practical work. In contrast, the flexible constraint algorithm proposed here is as costly as the hard constraint and unconstrained algorithms, but suffers from minor inaccuracy due to its non-conservation of energy, which makes it, for other reasons, also less suitable for practical work.
Figure 8.7: Bond-length distributions in liquid neopentane from 200 ps MD simulation at 273 K. Thick lines denote unconstrained, thin lines flexible constrained simulations. Solid lines: $p = 1 \text{kbar}$, dashed lines: $p = 5000 \text{kbar}$.

8.7 Acknowledgements

Special thanks to Dr. Tomas Hansson for interesting and helpful discussions. Financial support by the National Center of Competence in Research (NCCR) Structural Biology of the Swiss National Science Foundation (SNSF) is gratefully acknowledged.

8.8 Appendix

Here we present the equations for the exact flexible constraints algorithm. The SHAKE method determines the constraint force $f'_i$ on atom $i$ from

$$0 = \left( r'_k(t + \Delta t) + (\Delta t)^2 \left( m^{-1}_k f'_{ki}(t, \lambda_k) + m^{-1}_k f'_{ko}(t, \lambda_k) \right) \right)^2 - d'^2_k(t + \Delta t), \quad (8.22)$$
Figure 8.8: Radial distribution functions for liquid neopentane from 200 ps of MD simulation at 273 K. a) and b) are at $p = 1$ kbar, c) and d) at $p = 5000$ kbar. a) and c) show the radial distribution function of the central carbon atom with respect to all other central carbon atoms, b) and d) of the central carbon atom with respect to all the methyl groups in the system. The solid line denotes the unconstrained, the dotted line the hard constrained and the dashed line the flexible constrained system.

with the unconstrained or free flight positions $r_k^{nc}(t + \Delta t)$ denoted as $r_k^1(t + \Delta t)$ in the first iteration. For each following iteration $r_k^i(t + \Delta t)$ is updated towards the (final) constrained positions $r_k(t + \Delta t)$. The exact constraint forces due to constraint $k$ are given from Equations 8.13 and
8.21 as

\[ f_i^e(t) = \begin{array}{ll}
-2\lambda_k \left[ (\delta_{i,k_1} - \delta_{i,k_2}) r_k(t) - \frac{d_k(t + \Delta t)\mu_k}{K_k(\Delta t)^2} \right. \\
\left. \left( \nabla q_i \otimes \left( -m_{k_1}^{-1} (\Delta t)^2 \nabla q_{u_1} U_{nc}(q(t)) + m_{k_2}^{-1} (\Delta t)^2 \nabla q_{u_2} U_{nc}(q(t)) \right) \right) \frac{r_k^{nc}(t + \Delta t)}{|r_k^{nc}(t + \Delta t)|} + \\
\left( \nabla q_i \otimes r_k(t) \right) \left( \frac{r_k^{nc}(t + \Delta t)}{|r_k^{nc}(t + \Delta t)|} - \frac{r_k(t)}{|r_k(t)|} \right) \right] \\
i = 1, 2, \ldots, N. \quad (8.23)
\]

The positions are updated according to these forces and the calculation is iterated for all constraints until the change in the constraint forces per iteration approaches zero (because the change in \( \lambda_k \) per iteration is going to zero).

If we assume a potential energy term \( U_{nc} \) that consists of a sum of pair-wise interactions

\[ U_{nc}(q(t)) = \sum_{i,j=1}^{N} u(i,j) (q_i(t) - q_j(t)) = \sum_{i,j=1}^{N} u(r_{ij}(t)) = \sum_{i} \sum_{j>i} u_{ij}(t), \quad (8.24) \]

the second derivatives of an interaction term \( u_{ij} \) are

\[ \nabla q_i \otimes \nabla q_i u_{ij} = \nabla q_i \otimes \nabla q_j u_{ij} = -\nabla q_i \otimes \nabla q_i u_{ij} = -\nabla q_j \otimes \nabla q_i u_{ij}. \quad (8.25) \]

If we further assume that there is no interaction between the two atoms \( k_1 \) and \( k_2 \) forming constraint \( k \), the forces \( (f_k^e, f_k^t) \) of the constraint \( k \) on these two atoms can be written as

\[ f_i^e(t) = \begin{array}{ll}
-2\lambda_k \left[ (\delta_{i,k_1} - \delta_{i,k_2}) r_k(t) - \frac{d_k(t + \Delta t)\mu_k}{K_k(\Delta t)^2} \right. \\
\left. \left( \delta_{i,k_1} m_{k_1}^{-1} \nabla q_{u_1} \otimes \nabla q_{u_1} \sum_{j \neq k_1, k_2} u_{k_1,j}(t) + \delta_{i,k_2} m_{k_2}^{-1} \nabla q_{u_2} \otimes \nabla q_{u_2} \sum_{j \neq k_1, k_2} u_{k_2,j}(t) \right) \frac{r_k^{nc}(t + \Delta t)}{|r_k^{nc}(t + \Delta t)|} \right. \\
\left. + \left( \delta_{i,k_1} - \delta_{i,k_2} \right) \left( \frac{r_k^{nc}(t + \Delta t)}{|r_k^{nc}(t + \Delta t)|} - \frac{r_k(t)}{|r_k(t)|} \right) \right] \\
l = k_1, k_2. \quad (8.26)
\]

The forces from the constraint \( k \) on all other atoms \( j \neq k_1, k_2 \) are

\[ f_j^e(t) = \begin{array}{ll}
-2\lambda_k \left[ \frac{d_k(t + \Delta t)\mu_k}{K_k} \right. \\
\left. \left( \delta_{i,k_1} m_{k_1}^{-1} \nabla q_{u_1} \otimes \nabla q_{u_1} u_{k_1,j}(t) - m_{k_2}^{-1} \nabla q_{u_2} \otimes \nabla q_{u_2} u_{k_2,j}(t) \right) \frac{r_k^{nc}(t + \Delta t)}{|r_k^{nc}(t + \Delta t)|} \right] . \quad (8.27)
\]
Using these forces in Equation 8.15 yields

\[ 0 = \left( r_k(t + \Delta t) - \frac{d_k(t + \Delta t)}{K_k(\Delta t)^2} \right)^2 \]

\[ = \left[ \left( \frac{d_k(t + \Delta t)}{K_k(\Delta t)^2} \right) \right]^2 - \left( \frac{d_k(t + \Delta t)}{K_k(\Delta t)^2} \right)^2 \]

and after simplification

\[ 0 = \left( r_k(t + \Delta t) - 2 \frac{d_k(t + \Delta t)}{K_k(\Delta t)^2} \right)^2 \]

\[ + \left( \frac{m_{k_1}^{-1} \nabla q_{k_1} \otimes \nabla q_{k_1} \left( \sum_{j \neq k_1, k_2} u_{k_1,j}(t) \right) \left( \Delta t \right)^2 r_k^{nc}(t + \Delta t) \right) \]

\[ + \left( \frac{m_{k_2}^{-1} \nabla q_{k_2} \otimes \nabla q_{k_2} \left( \sum_{j \neq k_1, k_2} u_{k_2,j}(t) \right) \left( \Delta t \right)^2 r_k^{nc}(t + \Delta t) \right) \]

\[ - \left( d_k(t + \Delta t) \right)^2. \]
And finally solving for $\lambda_k$ and neglecting terms of second order in $\lambda_k$ results in

$$\lambda_k = \left( (d_k(t+\Delta t))^2 - \left( \frac{d_k(t+\Delta t)}{r_k(t+\Delta t)} \right)^2 \right)$$

$$\left[ -4(\Delta t)^2 r_k(t+\Delta t) \cdot \left( m_{k1}^{-1} + m_{k2}^{-1} \right) \right.$$

$$\left( r_k(t) - \frac{d_k(t+\Delta t)\mu_k}{K_k(\Delta t)^2} \left( \frac{\mathbf{v}_{nc}^k(t+\Delta t)}{|\mathbf{r}_k(t+\Delta t)|} - \frac{\mathbf{r}_k(t)}{|\mathbf{r}_k(t)|} \right) \right)$$

$$+ \frac{d_k(t+\Delta t)\mu_k}{K_k} \frac{\mathbf{v}_{nc}^k(t+\Delta t)}{|\mathbf{v}_{nc}^k(t+\Delta t)|}$$

$$\left( m_{k1}^{-2} \nabla q_{k1} \otimes \nabla q_{k1} \sum_{j \neq k_1, k_2} u_{k_1,j}(t) + m_{k2}^{-2} \nabla q_{k2} \otimes \nabla q_{k2} \sum_{j \neq k_1, k_2} u_{k_2,j}(t) \right)^{-1} (8.30)$$

We obtain the following exact flexible constraints algorithm:

1. calculate (unconstrained) forces $\mathbf{V}_{q_i} u_{ij}$ and Hessian $\nabla q_i \otimes \nabla q_i u_{ij}$ from positions $q(t)$.

2. calculate free flight or unconstrained velocities $\mathbf{v}_{nc}^k(t+\Delta t/2)$ and positions $q_{nc}^k(t+\Delta t)$ via the leap-frog scheme.

3. update the constraint lengths $d_k(t)$ according to Equation 8.9.

4. while constraints are not satisfied,

   4.1 for all constraints:

   4.1.1 calculate $\lambda_k$ according to Equation 8.30.

   4.1.2 update the positions of the atoms involved in the constraint using the constraint forces given in Equation 8.26.

   4.1.3 update the positions of all other atoms using the constraint forces given in Equation 8.27.

5. back-calculate the velocities from the constrained positions.

6. continue simulation with $t = t + \Delta t$ from Step 1.

The approximate flexible constraints algorithm is obtained from the exact one by omitting the calculation of the Hessian in step 1, using Equation 8.16 in step 3 and omitting step 4.1.3 completely. If we want to achieve conservation of energy no velocity-dependent terms are allowed in the potential energy function. It is possible to use only the unconstrained forces to determine the
8.8. Appendix

flexible bond lengths, ignoring the velocity terms. This means that the constraint length will not change (become longer) due to spinning of a constraint. This results in

\[
d_k = \frac{\mu_k}{(\Delta t)^2 K_k} \left( |r_k - (\Delta t)^2 m_k^{-1} \nabla q_k U_{nc}(q) + (\Delta t)^2 m_k^{-1} \nabla q_{k2} U_{nc}(q)| - |r_k| + d_k^0 \right).
\]

The energy conservation of the free rotor, two colliding ethane molecules on a line and the two orthogonal colliding ethanes is shown in Table 8.6, last column.

<table>
<thead>
<tr>
<th>system</th>
<th>unconstrained</th>
<th>hard constrained</th>
<th>flexible constrained</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>start</td>
<td>end</td>
<td>start</td>
</tr>
<tr>
<td>free rotor</td>
<td>2.50584</td>
<td>2.50585</td>
<td>2.50425</td>
</tr>
<tr>
<td>loss</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
<td>0</td>
<td>-0.0015</td>
</tr>
<tr>
<td>linear collision</td>
<td>start</td>
<td>end</td>
<td>loss</td>
</tr>
<tr>
<td></td>
<td>30.0693</td>
<td>30.0693</td>
<td>0.0</td>
</tr>
<tr>
<td>loss</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
</tr>
<tr>
<td>%</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
<td>0</td>
<td>0.0420</td>
</tr>
<tr>
<td>orthogonal collision</td>
<td>start</td>
<td>end</td>
<td>loss</td>
</tr>
<tr>
<td></td>
<td>60.1304</td>
<td>60.1304</td>
<td>0.0</td>
</tr>
<tr>
<td>loss</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0527</td>
</tr>
<tr>
<td>%</td>
<td>0</td>
<td>0</td>
<td>0.0537</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
<td>0</td>
<td>-0.0004</td>
</tr>
</tbody>
</table>

Table 8.6: Energy conservation of the different algorithms in simple test cases. Three versions of the flexible constraint algorithm were tested. The approximate version based on Equation 8.8 and on 8.7, and the exact version only using Equation 8.7 for the flexible constraint forces. Test cases were a free rotor, a collision of two ethanes moving on a line towards each other at equal speed, and a collision of two ethanes perpendicular towards each other moving such that the second one collides with the lower united atom of the first ethane (see Figure 8.1). The values are in kJmol\(^{-1}\). The simulations were performed for a 2000 steps with a timestep-size of 0.001 ps.

Even if there are no velocity dependent terms in the potential energy function, the algorithm will lead to (small) velocities along the constraints during the simulation (whenever a constraint length changes). The work resulting from these velocities can be estimated by
\[ w_k = \frac{\mu_k}{\Delta t} \frac{d_k(t + \Delta t) - d_k(t)}{|r_k(t)|} \left( \frac{p_{k_1}(t + \Delta t/2) - p_{k_1}(t - \Delta t/2)}{m_{k_1}} - \frac{p_{k_2}(t + \Delta t/2) - p_{k_2}(t - \Delta t/2)}{m_{k_2}} \right) \cdot r_k(t) \quad (8.32) \]

From *Table 8.6* one can see that the work done while quasi-adiabatically changing the constraint lengths will be missing from the system.
8.9 Bibliography


Chapter 9

Free energy calculations using flexible-constrained, hard-constrained and non-constrained MD simulations

9.1 Summary

A comparison of different treatments of bond-stretching interaction in molecular dynamics simulation is presented. Relative free energies from simulations using rigid bonds maintained with the SHAKE algorithm, using partially rigid bonds maintained with a recently introduced flexible constraints algorithm, and using fully flexible bonds are compared in a multi-configurational thermodynamic integration calculation of changing liquid water into liquid methanol. The formula for the free energy change due to a changing flexible constraint in a flexible constraint simulation is derived. To allow for a more direct comparison between these three methods, three different models for water and methanol were used: a flexible model (simulated without constraints and with flexible constraints), a rigid model (simulated with standard hard constraints), and a third model (simulated with flexible constraints and standard hard constraints) in which the ideal or constrained bond lengths correspond to the average bond lengths obtained from a short simulation of the unconstrained flexible model. Comparison of the relative free energies obtained from these simulations shows that the particular treatment of the bonds is of minor influence, whereas the relative free energy difference and the barrier to be overcome in the alchemical change of water to methanol between the various models is sizeable.
9.2 Introduction

With the advance in computational power simulation of liquids and solutions of biomolecules has become feasible. Newton’s equations of motion for ten thousands of particles are integrated forward in time using finite difference algorithms such as the leap-frog one. This requires an integration time step \( \Delta t \) which is about an order of magnitude smaller than the shortest oscillation time period in the molecular system. Therefore, bond vibrations with a time period of about \( 10-30 \text{fs} \) usually limit the size of the time step, which is standardly chosen as \( 0.5 \text{fs} \). Elimination of these fast oscillations through the application of hard constraints, e.g. using the SHAKE algorithm, allows for a longer time step of about \( 2 \text{fs} \). Although hard constraints are likely to be a more faithful representation of the quantum-mechanical nature of the bond-stretching vibrations than a classical-mechanical harmonic (or quartic) oscillator, as used in non-constrained simulations, they do not allow for a change in bond lengths during a simulation. However, in some processes a change in average bond lengths, e.g. under the influence of pressure, may play an essential role and should therefore be possible in a simulation. To this end flexible-constraint algorithms have been proposed. Using these methods the bond-length distance constraints are (adiabatically) adjusted to their current minimum-energy lengths according to the total energy (or total potential energy) of the system at the current time (step). Generally, these methods require multiple energy evaluations at every time step, which makes them an order of magnitude more expensive than hard-constraint algorithms. However, using a reasonable approximation this disadvantage could be overcome and a fast flexible-constraint algorithm obtained.

One of the most important quantities that can be obtained from simulations are relative free energies. To predict ligand binding or (relative) stabilities of biomolecules their relative free energy has to be known. Over the last decades many methods to access these free energy differences by molecular dynamics simulations have been developed as the calculation of the absolute free energies of systems comprising more than a handful of degrees of freedom seems unfeasible. Here, the formulae to obtain free energy differences from flexible-constraint simulations in which bond-length parameters are changed between the two end states, are derived. As an example of their application, the influence of the treatment of bond lengths on the (excess) free energy difference between water and methanol was investigated. Both for water and methanol, two flexible and a rigid model were used in the calculations.

9.3 Method

The free energy difference between state \( A \) and state \( B \) of a system can be determined through multi-configurational thermodynamic integration:

\[
\Delta G_{BA} = \int_{\lambda=A}^{B} \left< \frac{\partial H(p, r; \lambda)}{\partial \lambda} \right>_{\lambda} d\lambda,
\]

(9.1)
9.3. Method

\[ \mathcal{H}(p, r; \lambda) = \mathcal{K}(p; \lambda) + V(r; \lambda) \]

is the Hamiltonian of the system with \( \mathcal{K}(p; \lambda) = \sum_{i=1}^{N} \frac{p_i^2}{2m_i(\lambda)} \)

the kinetic energy and \( V(r; \lambda) \) an interaction energy term representing the interactions between the particles, \( r \equiv (r_1, r_2, \ldots, r_N) \) is indicating the \( N \)-particle configuration with \( r_i \) the position, \( p_i \) the momentum and \( m_i \) the mass of particle \( i \), and \( \lambda \) is a coupling parameter (alchemically) connecting state \( A \) and state \( B \). If one is interested in the excess free energy, then only the second term in the Hamiltonian representing the interactions between the particles is used,

\[
\Delta G_{BA}^{\text{exc}} = \int_{\lambda=A}^{B} \left\langle \frac{\partial V'(r; \lambda)}{\partial \lambda} \right\rangle d\lambda. \tag{9.2}
\]

Often, it is the relative excess free energy of two states which is experimentally available.

To compare the influence of different treatments of bonds on the free energy, the change of liquid water into liquid methanol using the coupling parameter approach was simulated with the GROMOS\textsuperscript{13–15} software for two flexible and a rigid model of water and of methanol. The contributions to the \( \lambda \)-derivative of the Hamiltonian of the pairwise nonbonded interaction energy term\textsuperscript{13}

\[
q_{ij}^{mb}(r; \lambda) = \sum_{\text{pairs } i,j} \lambda^n q_{ij}^{mb}(r_{ij}; B; 1 - \lambda) + (1 - \lambda)^n q_{ij}^{mb}(r_{ij}; A; \lambda) \tag{9.3}
\]

with

\[
q_{ij}^{mb}(r_{ij}; X; \lambda') = \frac{1}{\alpha_{LJ}(i, j)(\lambda')^2 C_{126}^X(i, j) + r_{ij}^6} \left( \frac{C_{X}^X(i, j)}{\alpha_{LJ}(i, j)(\lambda')(X)^2 C_{126}^X(i, j) + r_{ij}^6} - C_6^X(i, j) \right) + \frac{q_i^X q_j^X}{4\pi \varepsilon_0 \varepsilon_1} \left( \left( \frac{1}{(\alpha_C(i, j)(\lambda')^2 + r_{ij}^2)^{3/2}} - \frac{1}{2} C_{rf} r_{ij}^2 \right) - \frac{1}{2} C_{rf} \right) \tag{9.4}
\]

where \( C_{12}^X(i, j) \), \( C_6^X(i, j) \), \( C_{12}^B(i, j) \), and \( C_6^B(i, j) \) are the Lennard-Jones parameters for state \( A \) and state \( B \) for the pair of particles \( i, j \), \( q_i^A \), \( q_i^B \), \( q_j^A \), and \( q_j^B \) the respective charges and \( \alpha_{LJ}(i, j) \) and \( \alpha_C(i, j) \) the Lennard-Jones and Coulomb soft-core interaction function parameters, and

\[
C_{126}^X(i, j) = \begin{cases} 
C_{12}^X(i, j)/C_6^X(i, j) & \text{if } C_6^X(i, j) \neq 0 \\
0 & \text{otherwise}, \end{cases} \tag{9.5}
\]

is given by

\[
\frac{\partial q_{ij}^{mb}}{\partial \lambda}(r; \lambda) = \sum_{\text{pairs } i,j} n \lambda^{n-1} q_{ij}^{mb}(r_{ij}; B; 1 - \lambda) + \lambda^n \frac{\partial q_{ij}^{mb}(r_{ij}; B; 1 - \lambda)}{\partial \lambda} - n(1 - \lambda)^{n-1} q_{ij}^{mb}(r_{ij}; A; \lambda) + (1 - \lambda)^n \frac{\partial q_{ij}^{mb}(r_{ij}; A; \lambda)}{\partial \lambda}. \tag{9.6}
\]
with
\[
\frac{\partial \mathcal{V}_{ij}^{\text{mb}}(r_{ij};X';\lambda')}{\partial \lambda} = \frac{-2\lambda' \frac{\partial \lambda'}{\partial \lambda} \alpha_{ij}^{X}(i,j) C_{126}^{X}(i,j)}{\left( \alpha_{ij}^{X}(i,j) (\lambda')^2 C_{126}^{X}(i,j) + r_{ij}^{6} \right)^2} \\
\frac{2 C_{126}^{X}(i,j)}{\left( \alpha_{ij}^{X}(i,j) (\lambda')^2 C_{126}^{X}(i,j) + r_{ij}^{6} \right)} - C_{6}^{X}(i,j) - \\
\frac{q_{i}^{X} q_{j}^{X}}{4 \pi \varepsilon_{0} \varepsilon_{1}} \frac{\partial \lambda'}{\partial \lambda} \alpha_{C}(i,j) \\
\frac{1}{\left( \alpha_{C}(i,j) (\lambda')^2 + r_{ij}^{6} \right)^{3/2}} - \frac{3/2}{\left( \alpha_{C}(i,j) (\lambda')^2 + R_{ij}^{2} \right)^{5/2}} \quad (9.7)
\]
and with
\[
\frac{\partial \lambda'}{\partial \lambda} = \left\{ \begin{array}{ll}
1 & \text{for} \quad \lambda' = \lambda \\
-1 & \text{for} \quad \lambda' = (1 - \lambda). \end{array} \right. \quad (9.8)
\]

The contribution from the \( \lambda \)-dependent quartic bond potential energy term \(^{13}\)
\[
q_{\text{bond}}^{X}(r;\lambda) = \sum_{n=1}^{N_{b}} \frac{1}{4} \left( (1 - \lambda) K_{b}^{A} + \lambda K_{b}^{B} \right) \left( b_{n}(t) \right)^{2} - \left( (1 - \lambda) b_{0}^{A} + \lambda b_{0}^{B} \right)^{2} \quad (9.9)
\]
is
\[
\frac{\partial q_{\text{bond}}^{X}(r;\lambda)}{\partial \lambda} = \sum_{n=1}^{N_{b}} \frac{1}{4} \left[ -4 \left( K_{b}^{A} + \lambda (K_{b}^{B} - K_{b}^{A}) \right) \left( b_{0}^{B} - b_{0}^{A} \right) \\
\left( b_{0}^{B} + \lambda (b_{0}^{B} - b_{0}^{A}) \right) \left( b_{n}(t) \right)^{2} - \left( b_{0}^{A} + \lambda (b_{0}^{B} - b_{0}^{A}) \right)^{2} \right. \\
\left. \left( K_{b}^{B} - K_{b}^{A} \right) \left( b_{n}(t) \right)^{2} - \left( b_{0}^{A} + \lambda (b_{0}^{B} - b_{0}^{A}) \right)^{2} \right] \quad (9.10)
\]
for \( N_{b} \) bonds with force constants \( K_{b}^{A} \) and \( K_{b}^{B} \) and bond lengths \( b_{0}^{A} \) and \( b_{0}^{B} \) for state \( A \) and state \( B \), respectively. The current bond length \( b_{n}(t) \) is \( |r_{b_{n}}(t)| = |r_{b_{1}b_{2}}(t)| = |r_{b_{1}}(t) - r_{b_{2}}(t)| \) for the bonded particles \( b_{1} \) and \( b_{2} \).

When \( N_{c} \) constraints are applied using Lagrange multipliers \( l_{k} \) in a simulation, these appear formally as parameters in the Hamiltonian,
\[
\mathcal{H}(r,p;\lambda) = \mathcal{K}(p;\lambda) + \mathcal{V}(r;\lambda) + \sum_{k=1}^{N_{c}} l_{k}(t;\lambda) \sigma_{k}(r;\lambda), \quad (9.11)
\]
with the constraint equations
\[
\sigma_{k}(r;\lambda) \equiv r_{k}^{2} - \left( r_{k}^{0}(\lambda) \right)^{2} = r_{k}^{2} - \left( (1 - \lambda) r_{k}^{0,A} + \lambda r_{k}^{0,B} \right)^{2} = 0, \quad (9.12)
\]
9.3. Method

where \( r_k = r_{k1} - r_{k2} \) and \( r^0_A \) and \( r^0_B \) are the constraint distances in state A and state B, respectively. Thus the (changing) constraints also contribute to the free energy derivative

\[
\frac{\partial}{\partial \lambda} \sum_{k=1}^{N_c} l_k(t;\lambda) \sigma_k(r;\lambda) = \sum_{k=1}^{N_c} \sigma_k(r;\lambda) \frac{\partial l_k(t;\lambda)}{\partial \lambda} + l_k(t;\lambda) \frac{\partial \sigma_k(r;\lambda)}{\partial \lambda} = -2 \sum_{k=1}^{N_c} l_k(t;\lambda) r^0_k(\lambda) \left( r_k^0 - r^0_A \right)
\]

(9.13)

A complete derivation is given elsewhere.

The recently introduced flexible constraint algorithm uses time-dependent constraint lengths. Adding an additional dependence on \( \lambda \) to these constraint lengths leads to the following constraint expression

\[
\sigma_k^f(r;\lambda) \equiv r_k(t)^2 - (r^f_k(t;\lambda))^2 = r_k(t)^2 \left( \frac{F_k(t;\lambda)}{K_k(\lambda)} + r^0_k(\lambda) \right)^2 = r_k(t)^2 \left( \frac{F_k(t;\lambda)}{(1-\lambda)K_k^A + \lambda K_k^B} + (1-\lambda)r^0_A + \lambda r^0_B \right)^2 = 0, \tag{9.14}
\]

where \( r^f_k(t;\lambda) \) is the (current) constraint distance under influence of the (external) force \( F_k(t;\lambda) \) on constraint \( k \) with force constants \( K_k^A \) and \( K_k^B \) accounting for the flexibility and the ideal constraint lengths \( r^0_A \) and \( r^0_B \) for states A and B, respectively. The external force on constraint \( k \) is given as

\[
F_k(t;\lambda) = \frac{\mu_k(\lambda)}{(\Delta t)^2} \left[ r_k(t) + \Delta t v^u_{k1}(t + \Delta t/2) - (\Delta t)^2 (m_{k1}(\lambda))^{-1} \frac{\partial}{\partial r_{k1}} V(r;\lambda) \right.
\]

\[
- \Delta t v^u_{k2}(t + \Delta t/2) + (\Delta t)^2 (m_{k2}(\lambda))^{-1} \frac{\partial}{\partial r_{k2}} V(r;\lambda) \left| \right. - \Delta t v_k(t - \Delta t/2) - \left| r_k(t) \right| \right).
\]

(9.15)

where \( \mu_k(\lambda) = m_{k1}(\lambda) m_{k2}(\lambda)/(m_{k1}(\lambda) + m_{k2}(\lambda)) \) and the superscript "uc" indicates quantities resulting from an unconstrained step. The masses \( m_{k1}(\lambda) \) and \( m_{k2}(\lambda) \) are defined as \( m_k(\lambda) = (1-\lambda)m_k^A + \lambda m_k^B \). The contribution to the \( \lambda \)-derivative of the Hamiltonian can be calculated (using Equation 9.14) as

\[
\frac{\partial}{\partial \lambda} \sum_{k=1}^{N_c} l_k(t;\lambda) \sigma_k^f(r;\lambda) = \sum_{k=1}^{N_c} \sigma_k^f(r;\lambda) \frac{\partial l_k(t;\lambda)}{\partial \lambda} + l_k(t;\lambda) \frac{\partial \sigma_k^f(r;\lambda)}{\partial \lambda} = -2 \sum_{k=1}^{N_c} l_k(t;\lambda) r^0_k(\lambda) \left( \frac{\partial F_k(t;\lambda)}{\partial \lambda} K_k(\lambda) + \frac{\partial}{\partial \lambda} r^0_k(\lambda) \right), \tag{9.16}
\]
Chapter 9. Free energy calculations using flexible constraints

with

$$\frac{\partial}{\partial \lambda} f_k^0(\lambda) = r_k^{0B} - r_k^{0A},$$  \hspace{1cm} (9.17)$$

and

$$\frac{\partial}{\partial \lambda} F_k(t; \lambda) = -F_k(t; \lambda) \frac{K_k^B - K_k^A}{(K_k(\lambda))^2} + (K_k(\lambda))^{-1} \frac{\partial}{\partial \lambda} F_k(t; \lambda).$$  \hspace{1cm} (9.18)$$

As the external force $F_k(t; \lambda)$ might arise from an interaction energy function which is by itself dependent on $\lambda$ it may have a non-zero ($\lambda$-) derivative. Here, we only consider the case of the external force being completely determined by the nonbonded interaction function $q_{nb}(r; \lambda)$, but it is straightforward to add contributions of other interaction functions to the $\lambda$-derivative. We get

\[
\frac{\partial}{\partial \lambda} F_k(t; \lambda) = \frac{\mu_k(\lambda)}{(\Delta t)^2} \frac{\partial}{\partial \lambda} \left[ \left| r_k(t) + \Delta t v_{k1}^{\mu}(t + \Delta t / 2) - (\Delta t)^2 (m_k^2(\lambda))^{-1} \frac{\partial}{\partial r_{k1}} q_{nb}(r; \lambda) \right| - \right]
\]

\[
\Delta t v_{k2}^{\mu}(t + \Delta t / 2) + (\Delta t)^2 (m_k^2(\lambda))^{-1} \frac{\partial}{\partial r_{k2}} q_{nb}(r; \lambda) \right) + \frac{F_k(t; \lambda)}{(\Delta t)^2} \frac{\partial}{\partial \mu_k(\lambda)} \mu_k(\lambda)
\]

\[
(9.19)
\]

with

\[
\frac{\partial}{\partial \lambda} \mu_k(\lambda) = (\mu_k(\lambda))^2 \left( \frac{m_k^B - m_k^A}{(m_k^1(\lambda))^2} + \frac{m_k^B - m_k^A}{(m_k^2(\lambda))^2} \right).
\]

(9.20)

The derivative of the length of the unconstrained (free-flight) position

\[
\frac{\partial}{\partial \lambda} |r_k^{\mu}(t + \Delta t)| = \frac{\partial}{\partial \lambda} \left| r_k(t) + \Delta t v_{k1}^{\mu}(t + \Delta t / 2) - (\Delta t)^2 (m_k^2(\lambda))^{-1} \frac{\partial}{\partial r_{k1}} q_{nb}(r; \lambda) \right| - \]

\[
\Delta t v_{k2}^{\mu}(t + \Delta t / 2) + (\Delta t)^2 (m_k^2(\lambda))^{-1} \frac{\partial}{\partial r_{k2}} q_{nb}(r; \lambda) \right) + \frac{F_k(t; \lambda)}{(\Delta t)^2} \frac{\partial}{\partial \mu_k(\lambda)} \mu_k(\lambda)
\]

(9.21)

is obtained from the square root of the sum of the squared components ($\alpha = x, y, z$) of this (unconstrained position) vector

\[
\frac{\partial}{\partial \lambda} |r_k^{\mu}(t + \Delta t)| = \frac{\partial}{\partial \lambda} \sqrt{\sum_{\alpha=x,y,z} \left( r_{k,\alpha}^{\mu}(t + \Delta t) \right)^2}.
\]

(9.22)

Carrying out the derivation leads to

\[
\frac{\partial}{\partial \lambda} |r_k^{\mu}(t + \Delta t)| = \frac{1}{|r_k^{\mu}(t + \Delta t)|} \left[ \sum_{\alpha=x,y,z} (\Delta t)^2 \left( r_{k,\alpha}^{\mu}(t + \Delta t) \right) \right.
\]

\[
+ \frac{f_{k1,\alpha}^{nb}(m_k^1(\lambda))^{-1} + (m_k^2(\lambda))^{-1} \frac{\partial}{\partial \lambda} f_{k1,\alpha}^{nb}}{-f_{k2,\alpha}^{nb}(m_k^2(\lambda))^{-1} + (m_k^2(\lambda))^{-1} \frac{\partial}{\partial \lambda} f_{k2,\alpha}^{nb}}
\]

(9.23)
where $\mathbf{f}^{nb}_i = -\frac{\partial}{\partial \mathbf{r}_i} q^{nb}(r; \lambda)$ was used and with

$$\frac{\partial}{\partial \lambda} (m_i(\lambda))^{-1} = \frac{\partial}{\partial \lambda} \left( (1 - \lambda)m_i^A + \lambda m_i^B \right)^{-1} = -\frac{1}{(m_i(\lambda))^2} \left( m_i^B - m_i^A \right). \quad (9.24)$$

The force from the pairwise nonbonded interaction function term (consisting of a Lennard-Jones, a Coulomb and a reaction-field term) resulting from the interaction between particles $i$ and $j$ (Equation 9.3) is

$$\mathbf{f}^{nb,ij}_i = -\left( \lambda^n \frac{\partial q^{nb}_{ij}(r_{ij}; B; 1 - \lambda)}{\partial \mathbf{r}_i} + (1 - \lambda)^n \frac{\partial q^{nb}_{ij}(r_{ij}; A; \lambda)}{\partial \mathbf{r}_i} \right), \quad (9.25)$$

with

$$\frac{\partial q^{nb}_{ij}(r_{ij}; X; \lambda')}{\partial \mathbf{r}_i} = \frac{-6r_{ij}^4}{\left( \alpha_{LJ(i,j)}(\lambda')^2C_{126}(i,j) + r_{ij}^6 \right)^2} \left( \frac{2C_2^X(i,j)}{(\alpha_{LJ(i,j)}(\lambda')^2C_{126}(i,j) + r_{ij}^6)} - \frac{C_6^X(i,j)}{r_{ij}} \right) \mathbf{r}_{ij} \frac{q^X_i q^X_j}{4\pi\varepsilon_0\varepsilon_1} \left( \frac{1}{\alpha_C(i,j)(\lambda')^2 + r_{ij}^2} \right)^{3/2} + \frac{C_{rf}(i,j)}{\left( \alpha_C(i,j)(\lambda')^2 + r_{ij}^2 + R_{rf}^2 \right)^{3/2}} \right) \mathbf{r}_{ij}. \quad (9.26)$$

This finally leads to the $\lambda$-derivative of the forces in Equation 9.23

$$\frac{\partial}{\partial \lambda} \mathbf{f}^{nb,ij}_i = -n\lambda^{-1} \frac{\partial q^{nb}_{ij}(r_{ij}; B; 1 - \lambda)}{\partial \mathbf{r}_i} - \lambda^n \frac{\partial q^{nb}_{ij}(r_{ij}; B; 1 - \lambda)}{\partial \mathbf{r}_i} + (1 - \lambda)^n \frac{\partial q^{nb}_{ij}(r_{ij}; A; \lambda)}{\partial \mathbf{r}_i} - (1 - \lambda)^n \frac{\partial q^{nb}_{ij}(r_{ij}; A; \lambda)}{\partial \mathbf{r}_i}, \quad (9.27)$$
Chapter 9. Free energy calculations using flexible constraints

with

\[
\frac{\partial}{\partial \lambda} \left( \frac{\partial \mathcal{Q}^{\text{hb}}_{ij}(\mathbf{r}_{ij}; X; \lambda')}{\partial \mathbf{r}_i} \right) = -6r_{ij}^4(-2) \left( \alpha_{LJ}(i, j)(\lambda')^2C_{126}(i, j) + r_{ij}^6 \right)^{-3} \\
\left( 2\lambda \frac{\partial \lambda'}{\partial \lambda} \alpha_{LJ}(i, j)C_{126}(i, j) \right) \\
\left( \frac{2C_{126}(i, j)}{(\alpha_{LJ}(i, j)(\lambda')^2C_{126}(i, j) + r_{ij}^6 - C_{0}^X(i, j))} \right) \mathbf{r}_{ij} - \\
\frac{6r_{ij}^4}{(\alpha_{LJ}(i, j)(\lambda')^2C_{126}(i, j) + r_{ij}^6)^2} \\
2C_{126}(i, j)(-1) \left( \alpha_{LJ}(i, j)(\lambda')^2C_{126}(i, j) + r_{ij}^6 \right)^{-2} \\
\left( \alpha_{LJ}(i, j)C_{126}(i, j)2\lambda \frac{\partial \lambda'}{\partial \lambda} \right) \mathbf{r}_{ij} - \\
\frac{q_i^X q_j^X}{4\pi\varepsilon_0\varepsilon_1} \left( \frac{1}{2} \left( \alpha_C(i, j)(\lambda')^2 + r_{ij}^2 \right)^{-5/2} \left( \alpha_C(i, j)2\lambda \frac{\partial \lambda'}{\partial \lambda} \right) + \\
C_{rf} \left( \frac{3}{2} \left( \alpha_C(i, j)(\lambda')^2 + R_{rf}^2 \right)^{-5/2} \left( \alpha_C(i, j)2\lambda \frac{\partial \lambda'}{\partial \lambda} \right) \right) \mathbf{r}_{ij} \right), \tag{9.28}
\]

and with

\[
\frac{\partial \lambda'}{\partial \lambda} = \begin{cases} 
1 & \text{for } \lambda' = \lambda, \\
-1 & \text{for } \lambda' = (1 - \lambda). 
\end{cases} \tag{9.29}
\]

Finally, the contribution from particle \( j \) is the opposite of the one from particle \( i \)

\[
\frac{\partial}{\partial \lambda} \frac{\partial \mathcal{Q}^{\text{hb}}_{ij}(\mathbf{r}_{ij}; X; \lambda')}{\partial \mathbf{r}_j} = -\frac{\partial}{\partial \lambda} \frac{\partial \mathcal{Q}^{\text{hb}}_{ij}(\mathbf{r}_{ij}; X; \lambda')}{\partial \mathbf{r}_i}. \tag{9.30}
\]

9.4 Molecular models and computational procedure

The difference in (excess) free energy of liquid water and liquid methanol was calculated using no constraints for the bond-lengths (unconstrained simulation, uc), hard constraints (hc) or flexible constraints (fc). To compare the three different methods three distinct models for water and methanol were used. The first models were the standard rigid models for the two liquids (models R). For water the SPC model\(^{18}\) and for methanol the B3 model\(^{19}\) were used. From those models, flexible ones were constructed, using standard (GROMOS 45A3\(^{20}\)) force constants (models FF). Finally, for direct comparison of the flexible-constraint and hard-constraint methods a third set of models (FR) was used where the constraint lengths correspond to the average distances measured from a short (10 ps) unconstrained simulation (simulation parameters as given below) using the
flexible (FF) models. To allow an easier comparison of the models, the H-O-H or CH₃-O-H bond-angle were replaced by an additional CH₃-H or H-H bond. The parameters for the water and methanol models are summarised in Table 9.1.

To determine the free energy difference between water and methanol using the different models, multi-configurational thermodynamic integration was used for the flexible model (FF) using unconstrained and flexible constrained MD, for the rigid model (R) using hard constrained MD and for the rigid model with distances according to the averages of the flexible model (FR) using hard constrained and flexible constrained MD simulations. A cubic box containing 1000 molecules was simulated using periodic boundary conditions at a temperature of 300K maintained by weak temperature coupling²² (τₚ = 0.1 ps) and at a pressure of 1 atm maintained by weak pressure coupling²² (τₚ = 0.5 ps, κₚ = 4.575 × 10⁻⁴ (kJmol⁻¹nm⁻³)⁻¹, using isotropic scaling of the coordinates). If required, constraints were enforced with the SHAKE algorithm⁴ or with the flexible constraints algorithm⁸ with a relative geometric tolerance of 0.0001. Nonbonded interactions were handled using a triple-range cutoff scheme²³. Within a short-range cutoff radius of 0.8 nm, the interactions were evaluated every time step based on a pair-list recalculated every five time steps. The intermediate-range interactions up to a long-range cutoff radius of 1.4 nm

### Table 9.1: Parameters of the three models for liquid water: standard rigid¹⁸ (SPC/R), flexible²¹ (SPC/FF) and flexible (SPC/FR) using ideal bond-length and angle distances corresponding to the average distances from a short simulation using the flexible model, and of the three models for liquid methanol: standard rigid¹⁹ (MeOH/R), flexible (MeOH/FF) and flexible (MeOH/FR) using ideal bond-length and angle distances corresponding to the average distances from a short simulation using the flexible model. Distances (d) in nm, force constants (K) in 10⁵ kJmol⁻¹ nm⁻², charges (q) in e, and Lennard-Jones parameters \((C_{12})^{\frac{1}{2}}\) in \(10^{-3} (kJmolnm^{-6})^{\frac{1}{2}}\) and \((C_{6})^{\frac{1}{2}}\) in \((kJmolnm^{-6})^{\frac{1}{2}}\). The Lennard-Jones parameters for hydrogens are zero.
were evaluated simultaneously with each pair-list update, and assumed constant in between. To
account for electrostatic interactions beyond the long-range cutoff radius, a reaction-field ap¬
proximation was applied, using a relative dielectric permittivity of 66.25.

For each relative free energy determination, simulations at 101 λ-values (evenly spaced from
0 to 1) were performed. It turned out that simultaneous growing of methyl groups out of hydro¬
gens throughout the system led very quickly to instability. Trying out different soft-core param¬
eters (from 0.1 to 1.0) α_LJ and (from 0.001 nm^2 to 0.1 nm^2) α_C, shifted the problem with respect
to λ but did not solve it. A possible remedy might be to introduce different dependences on λ
for selected groups of atoms. We implemented a quadratic dependence on λ with one adjustable
parameter θ for each group of atoms

\[ \lambda''(\theta) = \theta \lambda^2 + (1 - \theta) \lambda. \] (9.31)

When calculating (nonbonded) interactions, for each pair of atoms the corresponding \( \lambda'' \) is cal¬
culated and used in the energy and force calculation. The λ-derivative of the Hamiltonian is
calculated with respect to \( \lambda'' \) and multiplied by an additional factor,

\[ \frac{\partial V_{ij}^{nb}(r_{ij}';\lambda'')} \partial \lambda }{ \partial \lambda'' } = \frac{\partial V_{ij}^{nb}(r_{ij}';\lambda'')} \partial \lambda'' }{ \partial \lambda'' } = \frac{\partial V_{ij}^{nb}(r_{ij}';\lambda'')} \partial \lambda'' }{ \partial \lambda'' } \left( (2\lambda - 1)\theta (i, j) + 1 \right). \] (9.32)

Still, for the extreme case of simultaneously growing a thousand new Lennard-Jones particles
(hydrogen to methyl) evenly spaced throughout the system, the quadratic form of Equation 9.31
does not allow enough variation or the number of atom groups with different λ-dependence
required for a stable simulation is too large to be practical.

A second approach to vary λ per configuration on a wider λ-range turned out to be easier.
Instead of assuming identical soft-core parameters for all particles with changing interactions,
each was assigned a random value from a uniform distribution between 0.1 and 0.5 for α_LJ
and between 0.001 nm^2 and 0.01 nm^2 for α_C. Nevertheless, in order to achieve smooth changes
between the single λ points in the thermodynamic integration, many more than the usual number
of λ points (about 20) were required. In all our simulations, 101 λ points were used. We note that
it is in principle possible to close the gaps between the single, discrete λ points by slow-growth
simulation, thereby continuously and smoothly changing the λ value. But this procedure leads to
non-equilibrium simulation, with the actual configuration constantly lagging behind the current
λ value. This drawback is most severe when crossing steep energy barriers, which is exactly
necessary here.

9.5 Results

The derivative of the excess free energy difference with respect to λ is shown in Figure 9.1. The
upper half of the table shows the values obtained from changing water into methanol, whereas the
lower half shows the values obtained by the reverse process, changing methanol into water. There are significant differences in the convergence behaviour. Whereas the results for the forward (water to methanol) and backward (methanol to water) process for the R and the FR model are well within 1 kJ mol\(^{-1}\), the differences using the FF model are above 2 kJ mol\(^{-1}\).

The relative free energies and excess free energies of methanol with respect to water for the three models and different treatments of bonds calculated by numerical integration are shown in Tables 9.2 and 9.3.

<table>
<thead>
<tr>
<th>Model</th>
<th>uc</th>
<th>fc</th>
<th>hc</th>
</tr>
</thead>
<tbody>
<tr>
<td>H(_2)O (\rightarrow) MeOH</td>
<td></td>
<td></td>
<td>-4.14</td>
</tr>
<tr>
<td>R</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FF</td>
<td>-11.56</td>
<td>-5.60</td>
<td>-</td>
</tr>
<tr>
<td>FR</td>
<td>-</td>
<td>2.55</td>
<td>-1.80</td>
</tr>
<tr>
<td>H(_2)O (\leftarrow) MeOH</td>
<td></td>
<td></td>
<td>-5.04</td>
</tr>
<tr>
<td>R</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FF</td>
<td>-13.54</td>
<td>-7.62</td>
<td>-</td>
</tr>
<tr>
<td>FR</td>
<td>-</td>
<td>1.94</td>
<td>-2.36</td>
</tr>
</tbody>
</table>

Table 9.2: Relative free energy, \(\int \frac{\partial H}{\partial \lambda} \lambda \, d\lambda\), of methanol with respect to water using the rigid models (R), the flexible models (FF), or the third models (FR) with ideal or constraint distances corresponding to simulation averages of the flexible models, and using unconstrained (uc), flexible constrained (fc), or hard constrained (hc) simulation, obtained from multi-configurational thermodynamic integration using 101 \(\lambda\)-points, either starting from water (upper half) or from methanol (lower half). Values are given in kJ mol\(^{-1}\). Errors are estimated using extrapolation of block averages\(^{26}\).

The experimental value for the relative excess Gibbs free energy of methanol with respect to water, calculated\(^{27,28}\) from molar volumes of the vapor and the liquid\(^{29}\), is 6.2 kJ mol\(^{-1}\). This value is much better reproduced by the FR model than by the other two models. Using an identical bond treatment, the excess relative free energy changes between the models by about 2.5 to 8 kJ mol\(^{-1}\), whereas changing the bond treatment using an identical model only results in changes of up to 4 kJ mol\(^{-1}\).

In Figure 9.2 the densities during the thermodynamic integration simulations are shown. For all models and simulation methods the densities reached a minimum at intermediate \(\lambda\) values, when the particle cores are at maximum softness. This suggests that the Coulomb part of the nonbonded interaction function was too soft compared to the van der Waals part. In other words, the tendency of the particles to overlap with each other due to a soft Lennard-Jones interaction which would lead to an increase in the density was more than compensated by the reduced
Figure 9.1: Derivative of the excess Gibbs free energy difference, $< \partial V / \partial \lambda >_\lambda$, with respect to $\lambda$ during multiconfigurational thermodynamic integration of changing water ($\lambda = 0$) into methanol ($\lambda = 1$) using the flexible models (FF, black symbols), the rigid models (R, red symbols), and the third models (FR, blue symbols) with distances corresponding to averages of the flexible one. At each of the 101 points 10 ps of simulation time were used to average over. Unconstrained simulations are represented in the top panel, hard constrained ones in the middle panel, and flexible constrained ones in the bottom panel. Values from the alchemical change of water into methanol are represented by diamond symbols, values for the reverse change from methanol into water by square symbols.
9.6. Conclusion

<table>
<thead>
<tr>
<th>Model</th>
<th>uc</th>
<th>fc</th>
<th>hc</th>
</tr>
</thead>
<tbody>
<tr>
<td>H₂O → MeOH</td>
<td></td>
<td></td>
<td>2.70</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td>-</td>
<td>2.70</td>
</tr>
<tr>
<td>FF</td>
<td>-1.07</td>
<td>1.23</td>
<td>-</td>
</tr>
<tr>
<td>FR</td>
<td></td>
<td>9.36</td>
<td>5.04</td>
</tr>
<tr>
<td>H₂O ← MeOH</td>
<td></td>
<td></td>
<td>1.80</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td>-</td>
<td>1.80</td>
</tr>
<tr>
<td>FF</td>
<td>-3.13</td>
<td>-0.80</td>
<td>-</td>
</tr>
<tr>
<td>FR</td>
<td></td>
<td>8.76</td>
<td>4.48</td>
</tr>
</tbody>
</table>

Table 9.3: Relative excess free energy, \( \int < \frac{\partial V}{\partial \lambda} >_{\lambda} d\lambda \), of methanol with respect to water using the rigid models (R), the flexible models (FF), or the third models (FR) with ideal or constraint distances corresponding to simulation averages of the flexible models, and using unconstrained (uc), flexible constrained (fc), or hard constrained (hc) simulation, obtained from multi-configurational thermodynamic integration using 101 \( \lambda \)-points, either starting from water (upper half) or from methanol (lower half). Values are given in kJ mol\(^{-1}\). Errors are estimated using extrapolation of block averages\(^{26}\).

Looking at the densities of the different thermodynamic integration simulations shows clearly where the process changing water into methanol deviates from its reverse process of changing methanol into water. From this, it is obvious that results belonging to the FF model are not yet converged, the ones from the R model are much closer to convergence, but only the FR model seems to be really converged. This agrees with the differences found between the excess free energies of the forward and their reverse processes.

To obtain better results, simulations have to be prolonged for selected \( \lambda \) values, until the densities obtained at one \( \lambda \) value converge.

9.6 Conclusion

A comparison between different treatments of bond lengths with regard to their influence on relative free energies of liquid water and methanol was presented. The different bond-length treatments included flexible (unconstrained) bonds, rigid (hard constrained) bonds and flexible constrained bonds, in which the constrained bond length is adapted to the current environment. To make this comparison, a flexible model (FF) for water and one for methanol as well as a rigid model (R) for each of the two liquids were used. In addition to those, a third pair of models...
Figure 9.2: Densities during multiconfigurational thermodynamic integration of changing water ($\lambda + 0$) into methanol ($\lambda = 1$) using the flexible models (FF, black symbols), the rigid models (R, red symbols), and the third models (FR, blue symbols) with distances corresponding to averages of the flexible one. At each of the 101 points 10 ps of simulation time were used to average over. Unconstrained simulations are represented in the top panel, hard constrained ones in the middle panel, and flexible constrained ones in the bottom panel. Values from the alchemical change of water into methanol are represented by diamond symbols, values for the reverse change from methanol into water by square symbols.
Acknowledgements

Financial support by the National Center of Competence in Research (NCCR) Structural Biology of the Swiss National Science Foundation (SNSF) is gratefully acknowledged.
9.8 Bibliography


Chapter 10

Outlook

A long time has passed since the first molecular dynamics simulation in 1959\textsuperscript{1}. From being a curiosity in the beginning\textsuperscript{2}, simulation has become a valuable tool in increasing the understanding of atomistic processes\textsuperscript{3} and is reaching predictive powers. The appeal of classical molecular dynamics simulation is rooted in the balance it achieves between accuracy, system size and (sampling) time scale. These three factors together determine the computational cost of the simulation. With the tremendous increase of computational power over the last decades it stands to reason whether classical simulation still meets the demands of the researchers or whether other methods with a different balance between the three components will become favoured. Arguably most focus is on accuracy. Apart from being heavily dependent on the quality of the force field used, classical simulations are unable to treat electronic degrees of freedom. As could be shown recently, this sets a limit on the accuracy any force field may achieve\textsuperscript{4}. Adding electronic polarization in a mean-field approximation may help\textsuperscript{5}, but is still not the same as treating electronic degrees of freedom explicitly as in mixed quantum-classical or Car-Parrinello simulations (CPMD). Still, there are many areas where accuracy might not be the biggest problem but system size and sampling time are the issues, as in protein folding, protein or lipid aggregation studies or simulations of complex mechanisms in cells like transport through membranes. For these kind of system sizes or sampling times, the quality of current force fields is still unknown. It might well be that through assessing force-field quality especially in long time-scale simulations noticeable improvements are still possible. One important example of such a process is the current investigation of the stability of \textgreek{a}-helices versus \textgreek{b}-sheets of model peptides. Sometimes it might be very hard or even impossible to come up with unique force-field parameters which are truly transferable among the huge variety of biomolecules used in todays simulations. Restraining a simulation to reproduce known experimental data might yield better overall behaviour and therefore increase the likelihood of accurate predictions on yet unknown properties or parts of the systems under investigation. Structure predictions using incomplete experimental data might profit substantially from an algorithm as the one presented in Chapter 7 where restraining is combined with a method to enhance sampling. If the aim is to simulate even bigger systems for
still longer time molecular dynamics simulation may still be the method of choice. The emerging coarse-grained models promise to make simulations of systems with up to a million of particles for microseconds reality. Nevertheless, these models work best for large numbers of identical, simple molecules. Careful selection of the subset of degrees of freedom to retain in future models of more complex molecules like peptides is necessary. Techniques that combine the fine-grained with the coarse-grained world like the one shown in Chapter 5 may help to benefit from the immensely faster sampling of configurational space available for the coarse-grained models while still being able to profit from specific atomistic interactions of the fine-grained representation.

Computer simulation is mainly limited by the available computational resources. Even with the tremendous increase those have seen over the years its appetite is not nearly satisfied nor likely to be in the near future. With the stagnation in the increase of clock speeds of processors a new trend to enhance performance is reemerging: parallelization. Within the next years as many as eight computational cores might be running in a standard desktop machine, leaving room for up to 32 cores for double or even quadruple processor server systems. With this development parallelization of simulation algorithms will be more and more important. Techniques that profit from simultaneous simulation of multiple copies of a system like replica-exchange simulation will benefit immensely, not the least because of the ease of implementation of this technique. Nevertheless, classic simulation algorithms and existing data structures need to be reevaluated in terms of scalability to parallel environments. The high level of modularity and the encapsulation used in GROMOS05 described in Chapter 2, together with the very simple but still reasonably efficient MPI (distributed memory) and OpenMP (shared memory) parallelization, will hopefully provide a useful framework for future development.

*Man muss nicht alle Berge ebnen wollen.*

German proverb
10.1 Bibliography


Curriculum Vitae

Personal Data

Name: Markus Christen
Date of birth: March 29, 1975
Place of birth: Bülach, Switzerland
Citizenship: Bülach (ZH) and Wynau (BE), Switzerland

Education

1982 – 1988 Primary school in Bonstetten (ZH)
1988 – 1994 Kantonsschule Limmattal (Urdorf, ZH)
September 1994 Matura (Typus B)
1995 – 2001 Chemistry studies at the
Eidgenössische Technische Hochschule (ETH) in Zürich, Switzerland
with focus on Organic and Computational Chemistry
2000 Diploma thesis at the Institute of
Organic Chemistry, supervised by Prof. Dr. Andrea Vasella
Thesis title: *Synthese von C(5')-ethinylierten RNA-Phosphoramiditen und ihr Einbau in Ribonucleinsäuren*
2001 Graduation as *Dipl. Chem. ETH*
2001 Internship with Polypure SA, Norway for three months as researcher
2001 – 2006 Ph.D. studies at the Laboratory of Physical Chemistry of the
ETH Zürich, supervised by Prof. Dr. Wilfred F. van Gunsteren