Computation of solvent entropies from molecular dynamics simulations

Fakultät für Physik
Georg–August–Universität Göttingen

DIPLOMARBEIT
von

Friedemann Reinhard

Göttingen 2005

Contents

1 Introduction

Thanks to the ever increasing capabilities of nowaday's computer hardware computational methods have established themselves as the third major tool of physics – beside theory and experiment. This is particularly obvious in biophysics. Here "all-atom" simulations of biological macromolecules over a nanosecond timescale have become feasible and contributed significantly to the understanding of biological processes [6], [10], [42].
Answering the question of *what* happens on the molecular scale however is only part of their power. Moreover they can even provide researchers with the answer to the question *why* this happens by computing the driving forces of the process under study.

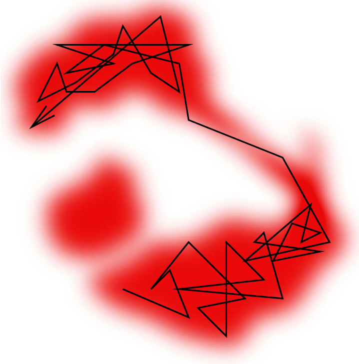

Fig. 1.1: An MD simulation yields a trajectory (black). For long simulation times, it will approximate an ensemble of the underlying density (red)

Under physiological conditions biological processes are subjected to constant pressure and temperature. The same environment is mimicked in computer simulations by applying thermostat and barostat algorithms [2] [36]. The simulation system therefore represents a classical statistical system whose state space is spanned by the coordinates $\vec{x}$ and momenta $\vec{p}$ of all atoms in the system. In a simulation Newton's equation of motion are solved yielding a trajectory $\vec{x}(t_i), \vec{p}(t_i)$. If the simulation is sufficiently long, the set of all $\vec{x}(t_i), \vec{p}(t_i)$ will approximate a thermodynamic ensemble, i.e., it will be distributed according to a Boltzmann density ρ (see Fig 1.1 for an illustration).

The driving forces of the system thus stem from the thermodynamic potential $G = pV + U - TS$, which is composed of pressure p, volume V, energy U, temperature T and entropy S. Since pV is nearly constant, we will omit this term throughout this thesis and replace G by the Helmholtz potential $F = U - TS$ to simplify notation.

The interplay of the constituents of G can be observed in many biological situations. A particularly famous one is the hydrophobic effect, which gives rise to various self-organization phenomena. The fact that lipids in aequous solution tend to stick together cannot be explained by their pairwise interaction: Lipids in vacuum do not attract each other. The formation of bilayers is mainly due to the entropy increase of the solvent. Single lipids impose high constraints on the water molecules around them thus lowering the entropy of the system. By self–assembling into a bilayer, these constraints on the water are relaxed. The water molecules can move freely again. The resulting increase in entropy favors the bilayer state. The computation of thermodynamical quantities like U or S in MD simulations is thus essential to answer the question above, namely why an observed process has happened.

In this work we focus on the calculation of entropies. These are defined as functionals of the phase space density $\rho(\vec{x})$, i.e, an integral over a function of ρ ranging over the whole phase space. This definition already shows what renders their computation extremely difficult. First, the integration over the whole phase space is computationally infeasible [29]. Second, the phase space density ρ is not directly accessible to a simulation. Rather, simulations yield only trajectories, i.e., series of points $\vec{x}(t_i)$ in configurational space. Assuming that the trajectory is long enough to approximate an ensemble of the density ρ, one can use it to calculate ensemble averages of local quantities like the energy. Since, however, the definition of entropy involves an ensemble average of the density itself, explicit knowledge of $\rho(\vec{x})$ is a prerequisite for its computation. One therefore has to estimate the density ρ from the trajectory $\vec{x}(t_i)$. If the estimated density has a sufficiently simple functional form, its entropy can be calculated analytically from the estimate.

This approach has been successfully established for the subspace spanned by the protein coordinates [24] [40]. A protein in aequous solution usually stays in its folded state during the whole simulation. In the phase space picture this means, that the ensemble elements $\vec{x}(t_i)$ generated by the MD simulation – regarded as points in 3N–dimensional space – cluster around a point $\vec{x}_0$ representing the average folded configuration. In this special case the density ρ is localized within a small region and can be sampled sufficiently even by a short simulation. A good approximation of this density can be obtained by fitting a multivariate Gaussian distribution to the trajectory. This particular fit function is motivated by the fact that the free energy landscape around the protein's average folded configuration will be dominated by a minimum of the free energy. If this minimum is approximately harmonic, it will give rise to a multivariate Gaussian distribution in the phase space region under study. This fit is routinely done in the analysis of MD simulations [39] [1].

As can be seen from the example of lipid bilayers, solvent entropies play a crucial role in biological processes. However, the solvent cannot be easily incorporated into the density estimation. The complications are twofold. First, the motion of the solvent molecules is diffusive, in contrast to the well localized atomic fluctuations in a protein. The configuration space that has to be sampled is therefore much too large to achieve convergence. Second, the typically very shallow energy landscapes generate phase space densities with quite complex topology, which precludes similarly straightforward analytical approximations.

The goal of this work is to address both problems by effectively "proteinizing" the solvent. The basic idea is to make use of its symmetry under permutations of the solvent molecules. For each element of the ensemble, the water molecules are relabeled such that the permuted configurations fall into a reasonably compact volume in phase space. To this aim we employ a combinatorial optimization algorithm [21] which relabels the water molecules such that their average distance to a predefined reference position becomes minimal.

While not changing any thermodynamic quantities of the ensemble, this approach should enhance sampling sufficiently to achieve convergence from relatively short simulations. It should also simplify the topology of the transformed density such that simple density estimates become applicable. The details of this approach are discussed in section 3.

In order to evaluate this idea we apply it to two simple yet typical systems, a free non–ideal gas and the solvation of a particle in this gas. To identify the strengths and weaknesses of our technique we choose a highly simplified gas model, namely a system of 2–dimensional repulsive disks moving in a plane.

As a more realistic test case we also apply our method to compute the entropy of a van–der–Waals gas and the solution entropy of an additional van–der–Waals particle in this gas. The technical details of these benchmarks are described in section 4.

Based on these results we suggest several possible ways to improve our density estimation. We examine whether existing algorithms might be able to refine the entropy estimate provided by our method. Here we focus on the technique of thermodynamic integration which yields precise values of entropy *differences* between two given systems. We investigate whether this method could be used in connection with our method which yields rough estimates of *absolute* entropies, i.e., the entropy of one isolated system. As a by–product of these considerations we propose an improved technique to compute energies from TI trajectories. Chapter 6.2 contains the details of these studies.

An alternative to finding a better fit function is to retain the Gaussian ansatz but transform the trajectory such that it fits to it. For this approach to work one has to find a transformation which leaves the entropy invariant. Such transformations are conveniently implemented by the motion of an incompressible fluid. As a proof of principle we apply this technique to a two dimensional test density. We draw an ensemble of 2D-points from it and move these points along the streamlines of an Eulerian fluid in order to transform them into a cloud of approximately Gaussian shape. The cloud's entropy is then estimated using the established Gauss-fit methods. This procedure should significantly enhance the entropy estimates. These proof–of–principle calculations are sketched in chapter 6.4.

In the last chapter we summarize our results and show how the proposed technique can be further enhanced. We also propose additional applications of the relabeling procedure.

2 Theory

2.1 Basic concepts

From the statistical physicist's point of view a simulation box is a set of N 3–dimensional coordinates $\vec{x}_a$ where N is the number of atoms in the box. This set can be considered as **one** point $\vec{x}$ in a $3N$–dimensional space. The $3N$–dimensional space of these coordinates will be referred to as **"configurational space"** below. This concept can be applied to subsystems of the box as well, e.g., the coordinates $\vec{x}_P$ and $\vec{x}_S$ of protein and solvent consisting of $3P$ and $3S$ dimensions respectively. The configurational space must not be confused with the **phase space**, which is the space of the atoms' coordinates and momenta $(\vec{p}, \vec{x})$.

In molecular dynamics (**MD**) simulations the motion of a system in configurational space is computed using classical mechanics. The atoms are assumed to be well described by the coordinates of their nuclei (Born–Oppenheimer approximation). The time development of these coordinates is computed by solving the Newton equations of motion arising from the Hamilton function

$$H(\vec{x}, \vec{p}) = \frac{\vec{p}^2}{2m} + V(\vec{x})$$

where $V(\vec{x})$ is the potential energy of the system describing the interaction of the different atoms. For this function several empiric models ("force fields") have been put forward. [22] [45]. This potential energy V is often visualized as a high dimensional **energy landscape** [35] [15].

An MD simulation yields a sequence of points $(\vec{x}(t_i), \vec{p}(t_i))$ in phase space. This sequence is generally referred to as a **"trajectory"**. Its points are called **configurations**. Forces $\vec{f}(t_i)$ and energies $E(t_i)$ are also computed in an MD simulation. The simulation system is coupled to a heat and/or a pressure bath. For long simulation times, the trajectory will thus approximate a statistical ensemble. Its data points $(\vec{x}(t_i), \vec{p}(t_i))$ will be distributed according to the canonical distribution [27]

$$\rho(\vec{x}) = \frac{e^{-\beta H(\vec{x},\vec{p})}}{Z} \tag{2.1}$$

where $\beta = 1/k_B T$ with Boltzmann's constant k_B and temperature T. Furthermore,

$$Z = \int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\, e^{-\beta H(\vec{x},\vec{p})}$$

is the partition function, where $\int_\Gamma$ denotes the integral over the whole phase space.

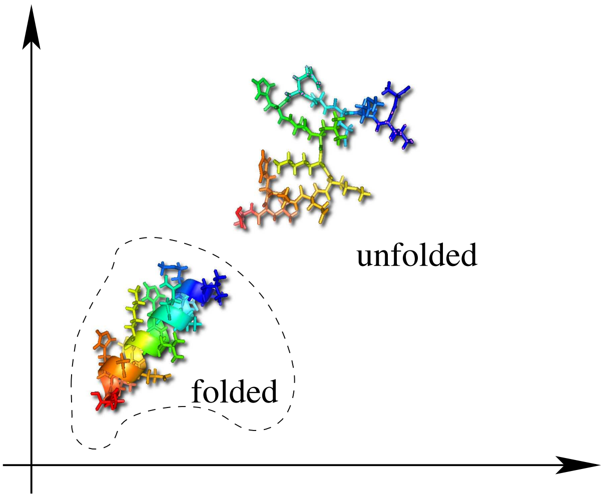

Fig. 2.1: Subdivision of the configuration space of a protein into folded and unfolded states. The probabilities of different states depend on their free energy.

2.2 Definition of statistical quantities

Only rarely are the coordinates $\vec{x}$ themselves the quantity of interest. More often one is interested in different "states" of the system such as the folded state of a protein. In the phase space picture (see Fig. 2.1) these states correspond to regions, i.e, to continuous sets of coordinates corresponding to a macroscopic property, e.g., a folded protein.

The stability of such states is governed by their respective probabilities. In the example of Fig. 2.1 one would consider the folded state as stable if the probability that, picking a random element of an ensemble, one finds the protein in the folded configuration is much higher than the probability for the unfolded case. This probability is given by

$$p_{\text{folded}} = \int_{\text{folded state}} d^{3N}\vec{x}\, d^{3N}\vec{p}\;\; \rho(\vec{x},\vec{p}) = \frac{\int_{\text{folded state}} d^{3N}\vec{x}\, d^{3N}\vec{p}\; e^{-\beta H(\vec{x},\vec{p})}}{Z} =: \frac{Z_{\text{folded}}}{Z}$$

Introducing the free energy

$$F = -k_B T \ln(Z) \tag{2.2}$$

the above stability criterion can be rewritten as

$$\Delta F = F_{\text{folded}} - F_{\text{unfolded}} \ll 0$$

where $F_{\text{folded}} = k_B T \ln(Z_{\text{folded}})$ and $F_{\text{unfolded}} = k_B T \ln(Z_{\text{unfolded}})$.

F can be decomposed into energy U and entropy S by means of the relation

$$\begin{aligned} F &= \int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\; H\; \rho(\vec{x},\vec{p}) + kT \int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\; \rho(\vec{x},\vec{p})\; \ln\rho(\vec{x},\vec{p}) \;= \\ &= U - TS \end{aligned} \tag{2.3}$$

The meaning of U is quite obvious: Parts of the phase space with low energy will be favored. Let us now turn to the interpretation of the second term, entropy:

2.3 Definition and interpretation of entropy

The entropy S is the subject of this work. Its definition follows from Eq. 2.3

$$S = -k_B \, \langle \ln \rho \rangle \tag{2.4}$$

where the functional $\langle \cdot \rangle = \int_\Gamma \cdot\, \rho(\vec{x},\vec{p})\;\; d^{3N}\vec{x}\, d^{3N}\vec{p}$ denotes the ensemble average. Due to this integration, also S is a functional mapping a given density $\rho(\vec{x},\vec{p})$ to a number S

In order to take the logarithm of the density it has to be dimensionless. This is accomplished by measuring the phase space in units of a reference volume. In order to be consistent with quantum statistical mechanics, this unit volume has to be chosen to be Planck's constant when one seeks to compute entropies of real systems. As long as one is only interested in the mathematical properties of the entropy, this factor can be omitted which we will do throughout this chapter.

The properties of S can be understood by considering $\ln(\rho)$ for the relevant domain $0 \leq \rho(\vec{x},\vec{p}) \leq 1$, as shown in Fig. 2.2. It is clear from this figure that large regions with low density will contribute much to the entropy whereas regions of high density, which have to be small due to the normalization of ρ, will yield smaller contributions. Therefore a large value of S implies that a large region of phase space is accessible to the system in a given state; a low value signifies that the system is constrained.

Phenomena like the hydrophobic effect suggest to split S into contributions from different subspaces of the system. Due to the nonlinearity of S it is in general impossible to do this in a physically meaningful way. Nevertheless such a decomposition is commonly assumed in the MD community.

Dividing the phase space into the subspaces Γ_P and Γ_S spanned by the protein and solvent coordinates and momenta $(\vec{x}_P, \vec{p}_P)$ and $(\vec{x}_S, \vec{p}_S)$ the probability density can be reduced to one subspace by integrating the full density over the other subspace. Thus

$$\rho_P(\vec{x}_P, \vec{p}_P) = \int_\Gamma d^{3N}x_S d^{3N}p_S\; \rho(\vec{x}_P, \vec{x}_S, \vec{p}_P, \vec{p}_S) \tag{2.5}$$

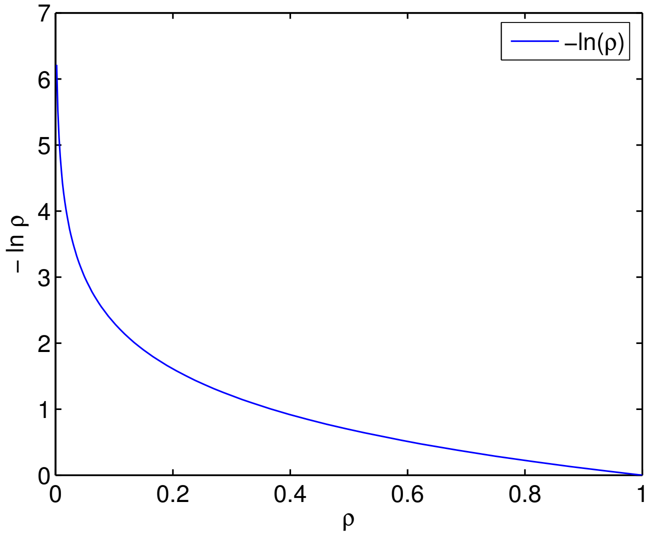

Fig. 2.2: $\ln(\rho)$ in the relevant domain. Phase space areas with a low density will contribute much to the entropy. The contribution of areas with a high density is low. The entropy will therefore be low for localized systems and high for systems which can access a large part of the phase space.

is the probability to find the system at the P subspace coordinate $\vec{x}_P$ regardless of its position in the S subspace.
The entropy of this density can be defined using a definition analogous to Eq. 2.4

$$S_P = -k_B \left\langle \ln(\rho_P(\vec{x}_P, \vec{p}_P)) \right\rangle_P$$

but its physical meaning is limited in the following sense: Two completely different densities can give rise to the same subspace entropy, even to the identical subspace density (see Fig 2.3 for a 2x1–dimensional illustration).

Knowledge of S_S and S_P is in general not sufficient to determine the entropy S of the full system. Only in the special case that the density factorizes,

$$\rho(\vec{x}, \vec{p}) = \rho_P(\vec{x}_P, \vec{p}_P)\rho_S(\vec{x}_S, \vec{p}_S), \tag{2.6}$$

S can be reconstructed from the subspace entropies by means of

$$S = S_P + S_S. \tag{2.7}$$

In the general case, correlations between the P and the S subspace may exist (See Fig. 2.4 for an illustration). In this case the density takes the form

$$\rho(\vec{x}, \vec{p}) = \rho_P(\vec{x}_P, \vec{p}_P)\rho_S(\vec{x}_S, \vec{p}_S)\rho_{\text{cor}}(\vec{x}_S, \vec{p}_S, \vec{x}_P, \vec{p}_P).$$

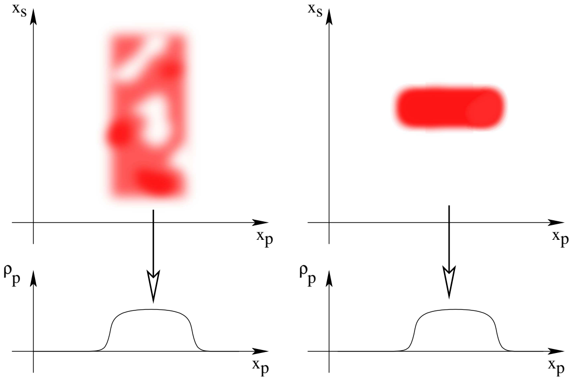

Fig. 2.3: Limited physical meaning of the subspace entropy. Two completely different densities with different total entropies can give rise to the same subspace density when they are projected by means of Eq. 2.5.

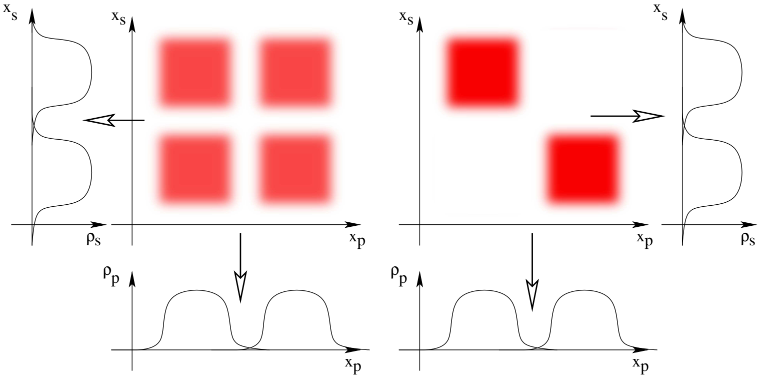

Fig. 2.4: Non–additivity of subspace entropies. Two completely different densities can give rise to the same subspace entropies, even to the same subspace densities. Knowledge of S_S and S_P is therefore not sufficient to determine the full entropy S.

Its entropy becomes thus

$$S = S_S + S_P + S_{\text{cor}}$$

Accordingly, entropy is additive only in the special case of Eq. 2.6.

In the biological context, these considerations imply that the solvent and protein contributions to the total entropy S cannot be split up in a physically meaningful way. Nevertheless, the protein subspace entropy is often computed. Assuming that the solvent contribution will be approximately the same for all proteins, this quantity can be used to compare entropies of proteins. This assumption needs justification and will likely often not hold.

Applying the above considerations to the subspaces of positions and momenta, however, a decomposition is possible. Since the only $\vec{p}$–dependent contribution to the Hamiltonian of a conservative system is

$$H_{\text{kin}} = \sum_{i=1}^{N} \frac{\vec{p}_i^2}{2m_i}, \tag{2.8}$$

the density factorizes into $\vec{p}$– and $\vec{x}$–dependent terms. The entropy, according to Eq. 2.7, is thus the sum of the $\vec{x}$– and $\vec{p}$–subspace entropies . The latter can be easily evaluated yielding a constant term.

2.4 Computation of the entropy

The aim of this work is to develop a method for the numerical computation of the entropy from a molecular dynamics or Monte–Carlo simulation. We are thus facing the problem: Given an ensemble $(\vec{x}(t_i), \vec{p}(t_i))$ distributed according to an unknown density $\rho(\vec{x}, \vec{p})$, estimate its entropy

$$S = -k_B \langle \ln(\rho) \rangle = -k_B \int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\, \rho(\vec{x}, \vec{p}) \ln\left[\rho(\vec{x}, \vec{p})\right] \tag{2.9}$$

In the following sections we will first review several existing methods tackling this problem, which we will use later on as building blocks and/or benchmarks for our approach.

It becomes clear from Eq. 2.9 that entropy cannot be computed in a straightforward way, because of two problems already mentioned in the introduction: It involves an integral over the whole phase space which is computationally infeasible and it requires knowledge of the density $\rho(\vec{x}, \vec{p})$ which is not accessible in a simulation.

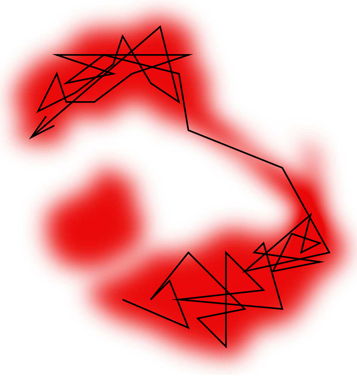

Fig. 2.5: density (red) versus trajectory (black)

There are two main ways to work around this difficulty: By means of the relation

$$\rho(\vec{x}, \vec{p}) = \frac{e^{-\beta H(\vec{x}, \vec{p})}}{Z}$$

one can reconstruct the density from the energy, which *can* be computed in the simulation. This approach however involves also the partition function Z which is difficult to obtain. Thermodynamic integration (chapter 2.4.2) is an entropy estimation method following this approach. However this method is only capable of calculating entropy differences.

Alternatively, a density can be obtained by fitting an analytical density ansatz to the trajectory. The following chapter describes the conventional technique of this kind.

2.4.1 Entropy Estimation using Principal–Component–Analysis

The most successful ansatz for a fit function was put forward by Karplus in 1981 [24]. It makes use of the fact that the entropy can be split into a position–dependent and a momentum–dependent part, as was explained above. Since the computation of the latter can be performed analytically the problem reduces to finding an ansatz for the configurational part of the density.

This is an extremely difficult task if one aims at an ansatz for the whole system comprising protein and solvent. For the protein subspace however a density estimation can be put forward, as was done by Karplus in his 1981 paper. Since all following derivations involve only the protein subspace, we will drop the label P and denote by $\vec{x}$ what was formerly $\vec{x}_P$. Also the number of degrees of freedom will be denoted by $3N$ instead of $3P$.

The Karplus ansatz for the density in the protein subspace is based on the following motivation:
One is mostly interested in the entropy of given states of the protein, such as the folded state. These states are often defined as the vicinity of a local or global minimum on the free energy landscape, i.e., the energy landscape including the protein energy as well as the mean contribution of interaction and solvent energy. In the vicinity of such a minimum $\vec{x}_0$, the free energy can be expanded to first order [14]

$$F \approx F(\vec{x}_0) + \frac{1}{2}(\vec{x} - \vec{x}_0)^T \boldsymbol{U} (\vec{x} - \vec{x}_0) \tag{2.10}$$

where $\boldsymbol{U} = \frac{\partial^2 F}{\partial x_i \partial x_j}|_{\vec{x}_0}$ is the second derivative of F at $\vec{x}_0$. The density near the minimum will thus be proportional to a multivariate Gaussian function

$$\rho(\vec{x}) \propto e^{-\frac{1}{2kT}(\vec{x}-\vec{x}_0)^T \boldsymbol{U} (\vec{x}-\vec{x}_0)} \tag{2.11}$$

Karplus suggested to obtain an estimation of the density by fitting a multivariate Gaussian function to the ensemble.

$$\rho(\vec{x}, \vec{p}) = \frac{1}{(2\pi)^{3N/2}|C|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{x}_0)^T \ \boldsymbol{C^{-1}} \ (\vec{x}-\vec{x}_0)} \tag{2.12}$$

Since this ansatz is motivated by Eq. 2.11, it is also referred to as **quasi-harmonic approximation**
Fit parameters of the ansatz are the function's center $\vec{x}_0$ and its $3N$ principal

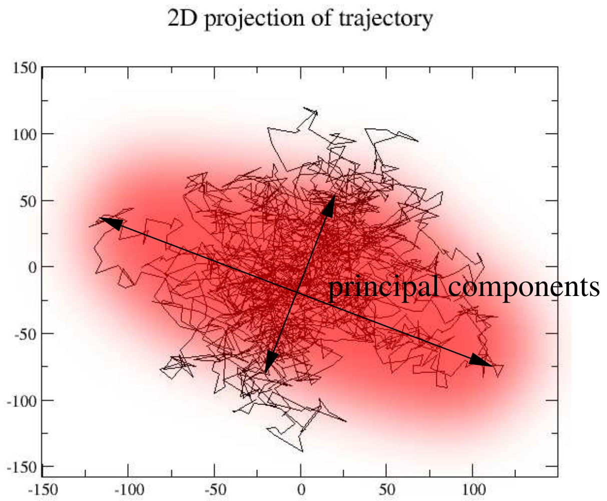

Fig. 2.6: Gaussian function fit to a trajectory. The center of the Gaussian function is set to the average configuration of the trajectory. The main axes of the Gaussian are chosen such that they equal the principal components of the trajectory.

components $\vec{a}_i$, which are the eigenvectors of the **covariance matrix** $\boldsymbol{C}$. $|C|$ denotes the determinant of $\boldsymbol{C}$.

Fitting this function to a trajectory is computationally inexpensive. The center $\vec{x}_0$ is obtained conveniently by averaging over the frames of the trajectory

$$\vec{x}_0 = \langle \vec{x}(t_i) \rangle_t$$

$\langle \cdot \rangle_t$ here denotes the time average, i.e, the average over all frames of the trajectory.

The main axes are chosen such that they equal the principal components of the trajectory. This is done by setting the covariance matrix C equal to the covariance matrix of the trajectory.

$$C_{ij} = \langle (x_i - x_{0i}) \cdot (x_j - x_{0j}) \rangle_t \tag{2.13}$$

The entropy of the fitted density can be computed analytically yielding

$$S = \frac{3}{2} N k_B + \frac{1}{2} k_B \ln \left[(2\pi)^{3N} |C| \right] \tag{2.14}$$

This approach however suffers from a technical problem: The covariance matrix of atomic fluctuations (Eq. 2.13) usually shows an eigenvalue spectrum as depicted in Fig 2.7. In particular it contains some small eigenvalues corresponding to high–frequency motions (like e.g., bond vibrations in the case of a protein). The determinant $|C|$ in Eq. 2.14 – being the product of all eigenvalues – thus becomes

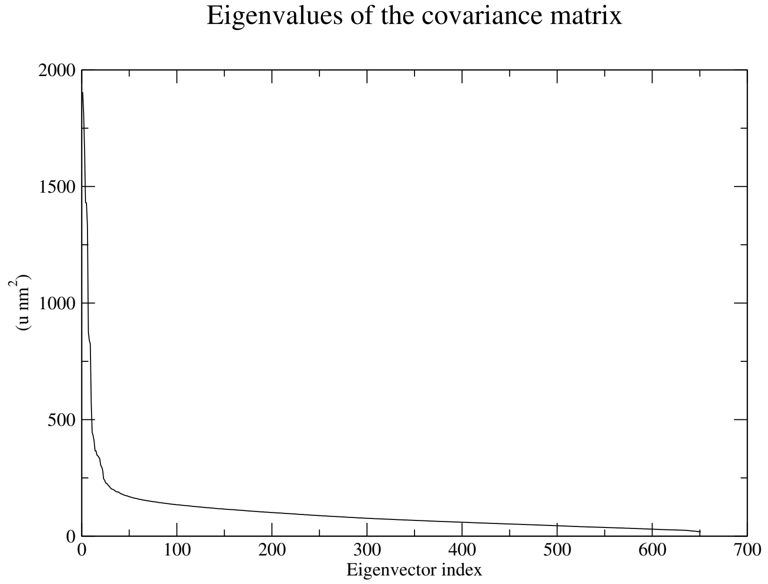

Fig. 2.7: typical eigenvalue spectrum (taken from a trajectory of a van–der–Waals gas)

very small and gives rise to negative absolute entropies. Entropy *differences* however are predicted correctly since in computing

$$\Delta S = S_2 - S_1 = \frac{k_B}{2} \ln(|C_2|/|C_1|)$$

the contributions of the small eigenvalues approximately cancel.

In 1993 J. Schlitter proposed a way to overcome the limitation and to extend the approach to the computation of absolute entropies [40]. He pointed out that the problematic high–frequency vibration modes correspond to fast motions like bond vibrations which are already in the quantum mechanical regime at room temperature. The density along these coordinates thus cannot be localized more sharply than the ground state wavefunction of the corresponding oscillator. Since the entropy of this ground state is zero, quantum mechanics imposes a lower limit for the entropy. Using the quantum mechanical formula for the entropy of a harmonic oscillator thus yields a natural way to cut off the low eigenvalues without disturbing the contributions of the low–frequency movements.

In his paper [40], Schlitter pointed out that in order to treat the multivariate harmonic oscillator quantum–mechanically, the $3N \times 3N$ mass–weighted covariance matrix

$$\boldsymbol{\sigma}_{ij} = \left\langle \sqrt{m_i m_j} x_i x_j \right\rangle$$

has to be considered instead of the pure covariance matrix $\boldsymbol{C} = \langle x_i x_j \rangle$. The quantum–mechanical entropy can be computed by the eigenvalues $\langle q_i^2 \rangle$ of this matrix by the relation

$$S \lesssim \frac{k_B}{2} \sum_{i=1}^{3N} \ln \left(1 + \frac{k_B T e^2}{\hbar^2} \left\langle q_i^2 \right\rangle \right) \tag{2.15}$$

This approach is now widely used to compute absolute entropies of proteins. Since however it still relies on the harmonic approximation, it is not suited for states of the protein comprising several distinct minima. For the same reason is not capable of handling the solvent, which is its main drawback.

2.4.2 Entropy Estimation using Thermodynamic Integration

Thermodynamic integration uses the relation between energy and phase–space density (Eq. 2.1) to compute the free energy or entropy difference between two systems [25] [46]. We will use it later as a benchmark for our entropy estimation method and therefore describe it in more detail here. Since free energies are more often computed than entropies we will sketch this case first.

The key quantity of statistical mechanics is the partition function Z.

$$Z = \int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\; e^{-\beta H(\vec{x},\vec{p})}$$

This cannot be estimated from an ensemble $(\vec{x}(t_i), \vec{p}(t_i))$, since the finite number of ensemble elements will in practice never cover the whole phase space. Accordingly, the same holds true for the free energy $F = -k_B T \ln Z$.

However, free energy *differences* between two slightly different systems can be computed. Let $H_1(\vec{x},\vec{p})$ and $H_2(\vec{x},\vec{p}) = H_1(\vec{x},\vec{p}) + \Delta H(\vec{x},\vec{p})$ be two slightly different Hamiltonians of the same coordinates and momenta. Then [44]

$$\Delta F \quad = \quad -k_B T \ln \left\langle e^{-\beta \Delta H} \right\rangle_{H_1} \tag{2.16}$$

where $\langle \cdot \rangle_{H_1}$ denotes the thermal average with respect to Hamiltonian H_1. This approach only works for small differences ΔH since configurations $(\vec{x},\vec{p})$ with a large negative $\Delta H(\vec{x},\vec{p})$ contribute much to ΔF but are unlikely to occur in an ensemble generated by H_1.

Free energy differences between two significantly different systems can be computed by splitting the difference Hamiltonian ΔH into n smaller parts $\Delta H_n = \Delta H/n$. This defines $n+1$ Hamiltonians “between” H_1 and H_2

$$H(i) = H_1 + \frac{i}{n} \Delta H$$

with $H(0) = H_1$ and $H(n) = H_2$. Their relative free energy differences $\Delta F_i = F_i - F_{i-1}$ can then be computed by Eq. 2.16.

Letting n tend to infinity one can define a continuous procedure to compute free energy differences. The Hamiltonians "between" H_1 and H_2 are then labeled by a continuous parameter $\lambda \in [0;1]$. A common choice is

$$H(\lambda) = (1-\lambda)H_1 + \lambda H_2 \tag{2.17}$$

Of course, other sufficiently smooth paths in the space of Hamiltonians with boundary conditions $H(0) = H_1$ and $H(1) = H_2$ are also possible.
In this limit Eq. 2.16 becomes

$$\left.\frac{\mathrm{d}F}{\mathrm{d}\lambda}\right|_{\lambda} = -k_B T \left\langle \left.\frac{\partial H}{\partial \lambda}\right|_{\lambda} \right\rangle_{H(\lambda)}$$

and the free energy difference between H_1 and H_2 is obtained by integrating over this result

$$\Delta F = -\int_0^1 \mathrm{d}\lambda \left\langle \left.\frac{\partial H}{\partial \lambda}\right|_{\lambda} \right\rangle_{H(\lambda)} \tag{2.18}$$

The pathway $H(\lambda)$ can be arbitrary, even unphysical. In practice, using Eq. 2.17, one often chooses it such that particles are created or annihilated by scaling their interaction Hamiltonians with λ.

Usually, λ is increased in every simulation step. Thus in every $H(\lambda)$ the system is sampled only one step long. The thermal average can thus be dropped which transforms Eq. 2.18 into the time average

$$\Delta F = -\int_0^T \mathrm{d}t \frac{\partial \lambda}{\partial t} \left.\frac{\partial H}{\partial \lambda}\right|_{\lambda}(t) = -\sum_{t_i} \mathrm{d}\lambda \left(\left.\frac{\partial H}{\partial \lambda}\right|_{\lambda} \right)(\vec{x}(t_i), \vec{p}(t_i))$$

where T is the run length of the simulation and $\mathrm{d}\lambda$ the increment of λ in a simulation step.

This method can be extended to the computation of *entropy* differences between two systems. The derivation is straightforward [37] [44]. By using the definition of the free energy (Eq. 2.3)

$$S = \frac{1}{T}(U-F) = \frac{\int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\, H(\vec{x},\vec{p}) e^{-\beta H(\vec{x},\vec{p})}}{T \int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\, e^{-\beta H(\vec{x},\vec{p})}} + k_B \ln \int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\, e^{-\beta H(\vec{x},\vec{p})}$$

one obtains the following expression for the entropy change

$$\left(\frac{\mathrm{d}S}{\mathrm{d}\lambda}\right)_{N,V,T} = \frac{1}{k_B T^2} \left[\left\langle \frac{\partial H(\lambda)}{\partial \lambda} \right\rangle_{H_\lambda} \langle H(\lambda) \rangle_{H_\lambda} - \left\langle \frac{\partial H(\lambda)}{\partial \lambda} H(\lambda) \right\rangle_{H(\lambda)} \right] \tag{2.19}$$

In this case λ cannot be increased in each step of the simulation, since the thermal averages cancel if they are taken only over one step. Instead, one has to perform multiple subsequent simulation runs at discrete values of λ. This is called **multistep perturbation**.

It should be noted that thermodynamic integration always computes free energy or entropy differences of the whole system, whereas the PCA–based entropy estimates are usually applied to *subsystems* like the subspace of protein coordinates, since only here the harmonic approximation can be justified.

Thermodynamic integration is not as common for the entropy as it is for the free energy, mainly because convergence is not as good as in the free–energy case. Nevertheless we will use it to benchmark and calibrate other techniques for the computation of entropies.

2.5 Analytical results

Later in this thesis we will apply the entropy estimates reviewed in the previous sections to several test cases and compare them with our method. For some of these test cases, analytical results can be derived. This is the aim of this section.

2.5.1 PCA of an ideal gas

This case is considered in [5] and [39] as a benchmark for the Schlitter method. Since the details are elaborated in both of these references, we will only recall their main results:
Consider an ideal gas of N non–interacting particles of mass m moving in a cubic box of sidelength L at temperature T. The variance of its density is computed to be

$$\langle x^2 \rangle = L^2/12$$

Inserting this result into Eq. 2.15 yields an entropy of

$$S = \frac{3N}{2} k_B \ln \left(1 + \frac{k_B T e^2}{\hbar^2} \frac{L^2}{12} \right) \tag{2.20}$$

In the case of an ideal gas, the entropy can also be computed analytically from the partition function. This approach yields the entropy [39]

$$S = N k_B \left(\ln \frac{V}{\lambda^3} + \frac{3}{2} \right) \tag{2.21}$$

with

$$\lambda = \frac{h}{\sqrt{2\pi m k_B T}}$$

being the thermal de Broglie wavelength. Note that in this derivation the particles are assumed to be distinguishable as stated explicitly in [39]. This assumption is neccessary since also the PCA is able to distinguish between the particles.

Interestingly, the Schlitter result is found to reproduce the analytical result almost exactly for large values of the box size. For a box of $V = 22.4$l [5] computes an error of less than 1%. This is surprising because the potential exerted by the box is far from harmonic as assumed by the harmonic approximation. The small error suggests that Gaussian fit functions can be applied to a wide range of anharmonic systems, too.

Since we will need it later, let us compare these two values explicitely for a box of 2.2 nm side length containing 100 particles of mass $m = 39.948$ amu at a temperature of $T = 300K$. Their de Broglie wavelength is

$$\lambda = \frac{h}{\sqrt{2\pi m k_B T}} = 1.5947 \cdot 10^{-11} m$$

so their exact entropy (Eq. 2.21) is

$$S_{\text{ana}} = 13.54 \frac{\text{kJ}}{\text{K mol}}$$

The Schlitter approach (Eq. 2.20) computes an entropy of

$$S_{\text{Schlitter}} = 13.98 \frac{\text{kJ}}{\text{K mol}}$$

The deviation from the analytical result is slightly larger than 3%.

2.5.2 Covariance matrix of a system of identical particles

In the case of a system of identical, but interacting particles, the covariance matrix cannot be computed fully analytically. However some of its properties can still be derived. In particular, its eigenvectors can be proven to belong to certain subspaces. Since we will study gases of interacting particles in this thesis, we will derive these properties in detail.

Consider a system of N identical interacting particles in d dimensions with position vectors $\vec{x}_i, i \in \{1, \dots N\}$ grouped together to one Nd–dimensional position vector $\vec{x}$. Let us decompose their covariance matrix $\vec{C} = \left\langle \vec{x}\vec{x}^T \right\rangle$ into N^2 $d \times d$ sub-matrices $C_{ij} = \left\langle \vec{x}_i \vec{x}_j^{\,T} \right\rangle$. Since the particles are indistinguishable only two different matrices will occur

$$C_{ij} = \begin{cases} \mathbf{C}_{=} \text{ for } i = j \\ \mathbf{C}_{\neq} \text{ for } i \neq j \end{cases}$$

The covariance matrix is thus of the form

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_= & \mathbf{C}_{\neq} & \mathbf{C}_{\neq} & \dots & \mathbf{C}_{\neq} \\ \mathbf{C}_{\neq} & \mathbf{C}_= & \mathbf{C}_{\neq} & \dots & \mathbf{C}_{\neq} \\ \vdots & & \ddots & & \vdots \\ \mathbf{C}_{\neq} & \mathbf{C}_{\neq} & \dots & & \mathbf{C}_= \end{pmatrix}$$

The eigenvalues and –vectors of such a matrix can also be computed analytically. Let $\mathbf{e}_+^k, \lambda_+^k$ be a set of d eigenvectors and –values of the matrix $\mathbf{C}_= + (N-1)\mathbf{C}_{\neq}$. Then clearly the vectors $\mathbf{E}_+^k = \left[\mathbf{e}_+^k, \mathbf{e}_+^k, \dots, \mathbf{e}_+^k\right]$ will be eigenvectors of the full covariance matrix, since

$$\mathbf{C}\mathbf{E}_+^k = \begin{pmatrix} (\mathbf{C}_= + (N-1)\mathbf{C}_{\neq}) \cdot \mathbf{e}_+^k \\ (\mathbf{C}_= + (N-1)\mathbf{C}_{\neq}) \cdot \mathbf{e}_+^k \\ \vdots \\ (\mathbf{C}_= + (N-1)\mathbf{C}_{\neq}) \cdot \mathbf{e}_+^k \end{pmatrix} = \lambda_+^k \mathbf{E}_+^k.$$

Let $\mathbf{e}_-^k, \lambda_-^k$ be eigenvectors and –values of the matrix $\mathbf{C}_= - \mathbf{C}_{\neq}$. Then for any pair $i \neq j$, the vector

$$\mathbf{E}_-^k = [0, 0, \dots, 0, \underbrace{\mathbf{e}_-^k}_{i}, 0, \dots, 0, \underbrace{-\mathbf{e}_-^k}_{j}, 0, \dots, 0]$$

is an eigenvector of $\mathbf{C}$, since

$$\mathbf{C}\mathbf{E}_-^k = \begin{pmatrix} 0 \\ \vdots \\ (\mathbf{C}_= - \mathbf{C}_{\neq}) \cdot \mathbf{e}_-^k \\ \vdots \\ -(\mathbf{C}_{\neq} - \mathbf{C}_=) \cdot \mathbf{e}_-^k \\ \vdots \\ 0 \end{pmatrix} = \lambda_-^k \mathbf{E}_-^k$$

The latter $dN(N-1)$ vectors belong to an eigenspace which is in fact only $(N-1)d$–dimensional. The eigenvalue spectrum of the covariance matrix will thus consist of the d center–of–mass–motion eigenvalues λ_+^k and the d eigenvalues λ_-^k of relative motion, each of which is $N-1$ times degenerate.

3 Computation of Solvent Entropy

3.1 Idea

As explained above, the density of a system can be estimated reliably from an ensemble if it stays close to one minimum of the energy surface. Then the quasi-harmonic approximation can be applied and the density can be obtained by fitting a multivariate Gaussian function to the ensemble generated by a molecular dynamics (MD) or Monte–Carlo (MC) simulation. As is shown by the ideal gas study, the quasi-harmonic approximation can also be applied to some anharmonic systems; it is definitely not applicable if the system samples several distinct minima of the energy surface.

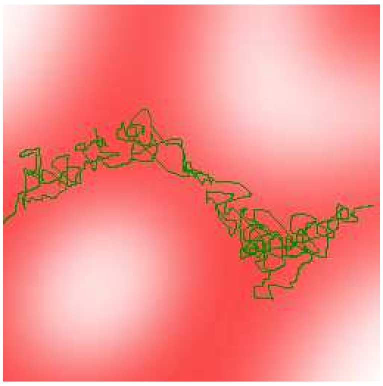

Fig. 3.1: schematic view of the solvent phase space density (shaded red) and trajectory (yellow). The energy landscape is shallow giving rise to diffusive motion and densities with complex topologies.

This however is exactly the case for the solvent environment in a simulation. Solvents like water move in a shallow energy landscape which generates an extended phase space density with many similar minima. Such a density renders the quasi-harmonic approximation inapplicable.

One might think about extending the approach by fitting more complicated model densities than the multivariate Gaussian to the ensemble. However we are not aware of a model density describing the whole phase space density of water. Furthermore, an unfeasibly lengthy simulation would be necessary to sample the whole phase space and adjust the necessarily many fitting parameters.

The aim of this thesis is to contribute to the solution of this problem in a complementary way. Instead of developing a complicated fit function for the water's phase space density, we propose to transform the water trajectory in a way that renders simpler fit functions applicable.

To this aim we exploit the symmetry of the water under permutations of the water molecules.
Let $\pi : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}$ be a permutation of the numbers $\{1, 2, \ldots, N\}$. These permutations form a group [23], which we will denote by Sym_N. Consider a system containing N water molecules. The 9 coordinates of the oxygen and the two hydrogen atoms

$$\vec{x}_{O,i}, \vec{x}_{H1,i}, \vec{x}_{H2,i}$$

of one water molecule will for simplicity be denoted by a 9-d vector $\vec{x}_i$. Let again $\vec{x}$ denote the $9N$–dimensional vector of all water molecules and $\vec{x}_\pi$ the same vector with the water indices permuted according to π, i.e.,

$$\vec{x}_\pi = [\vec{x}_{\pi(1)}, \vec{x}_{\pi(2)}, \ldots \vec{x}_{\pi(N)}]$$

The system's Hamiltonian is invariant under permutations of the molecules

$$H(\vec{x}, \vec{p}) = H(\vec{x}_\pi, \vec{p}_\pi) \tag{3.1}$$

if the molecules are indistinguishable. This is not the case if they are bound to each other or subject to an otherwise index–dependent force, which however is not the case for solvents like water.

Equation 3.1 has an important consequence. The solvent configurations $\vec{x}$ and $\vec{x}_\pi$ are equally likely to occur in a simulation, since

$$\rho(\vec{x}, \vec{p}) = \frac{e^{-\beta H(\vec{x},\vec{p})}}{Z} = \frac{e^{-\beta H(\vec{x}_\pi,\vec{p}_\pi)}}{Z} = \rho(\vec{x}_\pi, \vec{p}_\pi) \tag{3.2}$$

Conversely this implies that the water molecules may be relabeled (i.e., their coordinates may be swapped according to a permutation π) in every frame of the simulation without changing thermodynamical quantities like the mean energy.

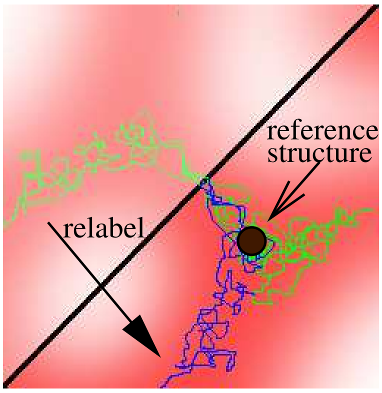

Fig. 3.2: Schematic view of the relabeling procedure in phase space. Solvent molecules will be relabeled, if the configuration can come closer to its reference structure in doing so.

We want to use this freedom and relabel the molecules according to some permutation π in every ensemble element such that the resulting ensemble can be fit by the methods of section 2.4.1. The crucial step for this approach to work is of course to find the appropriate permutations π.

Ideally, the permutations should transform all ensemble elements into the vicinity of a minimum of the potential energy surface, since then the quasi-harmonic approximation would become applicable. We propose to accomplish this by defining a reference position $\vec{x}_0$ lying in the vicinity of a minimum of the energy surface and finding permutations π such that the distance between the relabeled ensemble elements and the reference position becomes minimal. The relabeled ensemble will thus sample a smaller region of the phase space, the sampling density however will increase. In this way, the sampling problem should also be alleviated significantly.

Since MD simulations usually start from a configuration whose energy has been

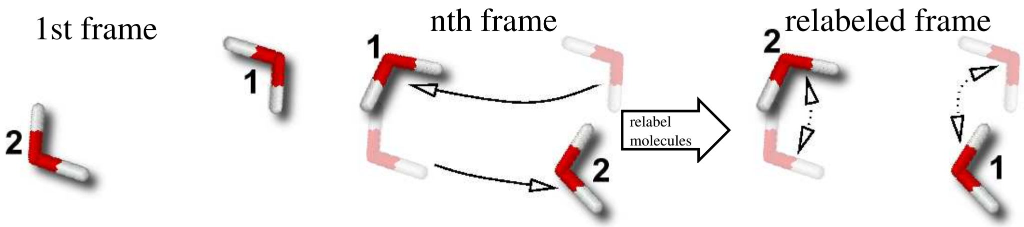

Fig. 3.3: illustration of the relabeling procedure in real space. If two molecules "change places", they can be brought closer to the reference structure by swapping their coordinates.

minimized, the first simulation frame provides a convenient choice for the reference position.

More formally, the following problem has to be solved for each ensemble element: Given two configurations $\vec{x}, \vec{x}_0 \in \mathcal{R}^{3N}$ find a permutation $\pi \in \mathrm{Sym}_N$ such that $\forall \sigma \in \mathrm{Sym}_N, \sigma \neq \pi$

$$(\vec{x}_\pi - \vec{x}_0)^2 \leq (\vec{x}_\sigma - \vec{x}_0)^2 \tag{3.3}$$

That is: find a permutation of the solvent molecules which brings the solvent as close to its reference position as possible. An illustration for the case of two molecules is given in Fig. 3.3.

3.2 The linear assignment problem

This task is a special case of a problem known in computer science as **linear sum assignment problem** (LSAP) or linear assignment problem (LAP). It is formulated as follows [8]: Given a $N \times N$ cost function c_{ij}, find the permutation $\pi \in \mathrm{Sym}_N$ maximizing the total cost

$$\sum_{i=1}^{N} c_{i\pi(i)}$$

Since the distance between permuted configuration and reference can be written as

$$(\vec{x}_\pi - \vec{x}_0)^2 = \sum_{i=1}^{N} (\vec{x}_{\pi(i)} - \vec{x}_{0i})^2$$

it becomes immediately clear that the condition of Eq. 3.3 is equivalent to the LSAP if we set

$$c_{ij} = -(\vec{x}_j - \vec{x}_{0i})^2. \tag{3.4}$$

Since the LAP occurs in many fields, there exist a number of efficient solutions. The earliest one is due to Kuhn [26] who realized that a polynomial solution of

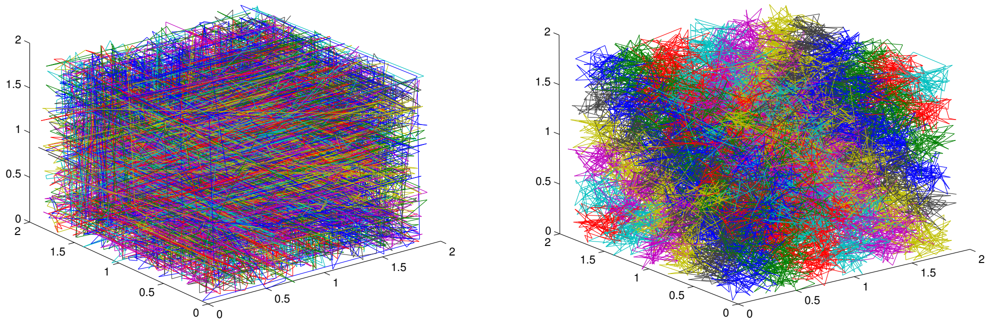

Fig. 3.4: Trajectories of the water oxygens without (left) and with (right) relabeling. Without relabeling (left), the molecules diffuse freely through the whole box volume. When the relabeling is switched on (right), they stay close to their reference position since they will be relabeled if they move too far.

the problem is possible and proposed an $\mathcal{O}(n^4)$ algorithm known today as the "Hungarian Method". Since then, different algorithms have been proposed, the best ones of which achieve $\mathcal{O}(n^3)$ efficiency. Most of them use linear optimization or primal–dual algorithms, techniques whose details we will omit here. They can be found in [8] [7] or [9].

3.3 The relabeled ensemble in real space

To gain more insight into the proposed method, let us consider a simulation of 216 water molecules moving in a box as an example. The data used for generating the Figures 3.3 and 3.5 was produced by an MD simulation, whose details will be described in section 4.

To visualize a trajectory of the system, we color the molecules differently and plot all configurations of the trajectory on top of each other. To get more transparent plots, we plotted only the oxygen coordinates. The resulting picture is shown in Fig 3.3.
The left part of the figure shows the trajectory before the relabeling is applied. The molecules diffuse through the box freely. The right part shows the same trajectory after we applied the relabeling. Here, the water molecules stay close to their reference positions. This is a nontrivial result: Criterion 3.3 considers only the distance in configurational space which involves all molecules. It is not obvious that this constrains also the movement of every single molecule. Since however moving a single molecule closer to its reference position will always reduce the distance of the whole ensemble, too, the method will apply such a permutation whenever it is possible. This is the case in almost every frame, since the high

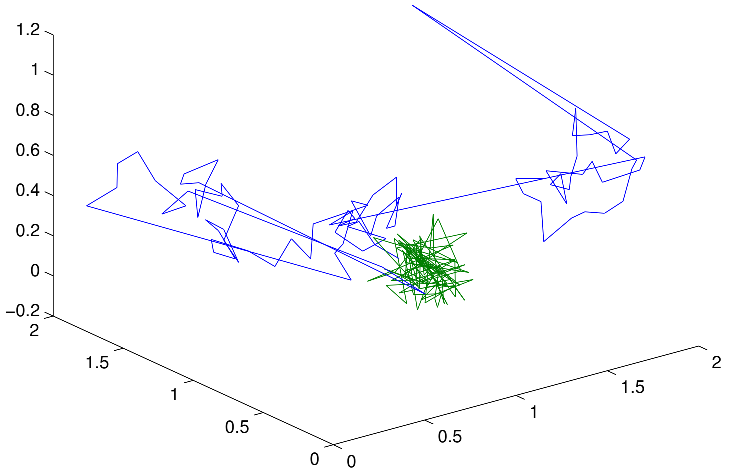

Fig. 3.5: Trajectories of a single water oxygen before and after the relabeling (blue and green curve respectively). The discontinuities are due to periodic boundary conditions.

density of the water makes it very probable that there will be at least one water molecule in the vicinity of each molecule's reference position. The size of the volume accessible to a single water molecule is of the order of the nearest neighbor distance.

Fig. 3.5 shows the same plot for only one molecule. Here the contrast between diffusive motion before and localized motion after the relabeling is even more striking.

3.4 The relabeled ensemble in configurational space

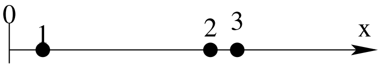

Fig. 3.6: 3x1–dimensional model system

Only part of the configurational space will remain accessible to the system when the relabeling is applied. Since we aim at an estimation of the entropy in this part, we have to address the question how probability densities look like in this part of the space. A prerequisite for this is to know as many details as possible about the shape of the part itself. The aim of this section is to provide these details by a characterization of the configurational space accessible to a system after the relabeling is applied.

Since the configuration space of real systems is too high–dimensional to be visualized, we start by studying a very low–dimensional example, namely 3 one–

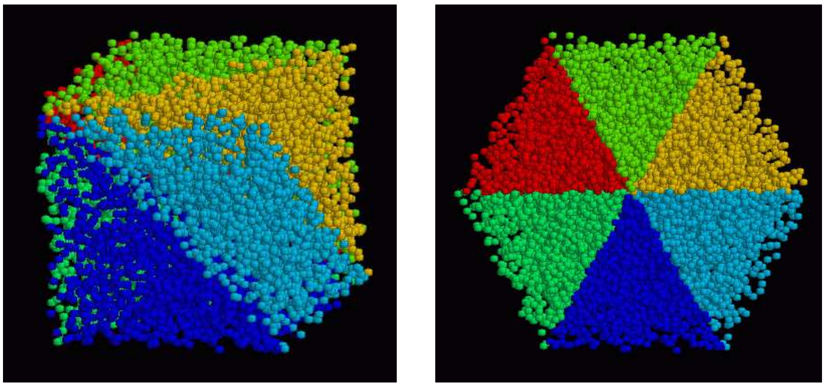

Fig. 3.7: Division of the 3x1D configurational space according to applied permutations. The right picture is taken from the (111)perspective. Switching on the relabeling would constrain the system in one of the colored slices.

dimensional particles (Fig. 3.4). Here a visualization is possible, since the configurational space is only 3–dimensional. The action of the relabeling algorithm can now be visualized by plotting all frames of an unrelabeled trajectory on top of each other and coloring them according to the permutation which the relabeling algorithm would apply. The resulting picture is shown in Fig. 3.7. As expected, the phase space is cut into 6 (= 3!) slices. Switching on the relabeling algorithm would condense the whole trajectory into the slice containing the reference position.

Some general features of the slice are already apparent in this plot: It is delimited by hyperplanes in the configurational space. Furthermore, its longest dimension is along the (111)–axis which is completely contained within the slice.

These features are general and can be proven. To do this, we introduce permutation matrices P defined by

$$P \cdot \vec{x} = \vec{x}_\pi.$$

That is, P swaps the coordinates according to π when applied to a vector $\vec{x}$. A permutation π will only be applied by the LAP solver, if it reduces the distance

$$(P\vec{x} - \vec{x}_0)^2 \leq (\vec{x} - \vec{x}_0)^2.$$

When both sides of the equation are expanded, only the mixed terms do not cancel and the condition reduces to

$$\vec{x_0}^T (1 - P)\vec{x} \leq 0$$

The phase space accessible to the relabeled trajectory is the set of all $\vec{x}$ for which no permutation reduces the distance, i.e., the set of all $\vec{x}$ for which $\forall P$

$$\vec{x_0}^T (1 - P)\vec{x} > 0 \tag{3.5}$$

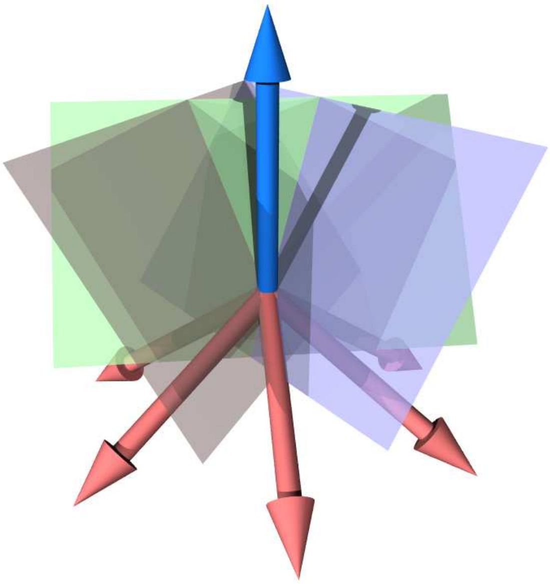

Fig. 3.8: Visualization of Eq 3.5. The starting point (vector $\vec{x}_0$, shown in blue) generates $N!$ vectors $\vec{x}^{0T}(1-P)$ (which are shaded red). They form the normal vectors of the planes delimiting the slice.

This can be viewed as $N!$ half-space equations in normal form, since $\vec{x_0}^T(1-P) = ((1-P)^T\vec{x}_0)^T$ are simply $N!$ vectors. Condition 3.5 states that the projection of $\vec{x}$ onto all of these vectors has to be positive, i.e., $\vec{x}$ has to lie in the intersection of these $N!$ positive half-spaces. The half-spaces are delimited by the planes defined by

$$\vec{x_0}^T(1-P)\vec{x} = 0.$$

See Fig. 3.4 for an illustration. Eq. 3.5 also allows to understand another characteristic property of the slice: Consider a vector $\vec{x} = [\vec{a}, \vec{a}, \ldots, \vec{a}]$, i.e., a configuration where all atoms are located at the same position. In this case $P \cdot \vec{x} = \vec{x}$, so condition 3.5 is automatically fulfilled.

Since this vector points along the largest dimension of phase space itself, it also represents the direction of largest dimension of the slice. It is however unlikely that the largest dimension of a relabeled trajectory will lie in this direction, since it corresponds to the center–of–mass–motion which is usually removed from a simulation.

3.5 Entropy of a relabeled density

Since the density of the relabeled system covers only a fraction of the configurational space, its entropy will be smaller than that of the unrelabeled one. To compare the entropy estimated from a relabeled trajectory to the entropy estimated from an unrelabeled trajectory, one has to take this difference into account. This can be done as follows:
As in section 2.5.1, we assume the particles to be distinguishable. Let $\vec{x}_s \in S$ lie inside the subset S of the configurational space which is accessible to the relabeled trajectory (the "slice"). $N!$ energetically equivalent configurations can be obtained by means of the relation $\vec{x} = P \cdot \vec{x}_s$. All these configurations will be mapped into the slice by the relabeling procedure, so the relabeling will increase the density in the slice by a factor of $N!$.

$$\rho_S(\vec{x}) = N!\rho(\vec{x}).$$

The whole configurational space Γ_C can be thought of as consisting of $N!$ identical copies of S so that an integral of an arbitrary function of the unrelabeled density $f(\rho)$ over S can be expressed as

$$\int_S d^{3N}x \; f(\rho(\vec{x})) = \frac{1}{N!}\int_{\Gamma_C} d^{3N}x f(\rho(\vec{x}))$$

Using these relations the configurational entropy S_S of the relabeled density is

$$\begin{aligned} S_S &= -k_B \int_S d^{3N}x \; \rho_S(\vec{x}) \ln\left[\rho_S(\vec{x})\right] = \\ &= -k_B \int_S d^{3N}x \; N! \; \rho(\vec{x}) \ln\left[N!\rho(\vec{x})\right] \\ &= -k_B \int_{\Gamma_C} d^{3N}x\rho(\vec{x}) \left[\ln\left[\rho(\vec{x})\right] + \ln N!\right] \\ &= S - k_B \ln(N!). \end{aligned} \tag{3.6}$$

Since the entropy in momentum space is not affected by the relabeling, this relation also holds true for the total entropy.

4 Methods

4.1 Our implementation

To evaluate the approach presented in the previous sections, we first had to implement it in a way that can be used together with the tools used to run and analyze MD simulations. We therefore developed a computer program called “g_permute”. It reads a trajectory in the standard formats used by the MD software GROMACS, solves the linear assignment problem, relabels the water molecules according to the solution of the LAP and outputs the relabeled trajectory in a standard format again. Details of its usage can be found in Appendix A.

The key step in the proposed method is to solve the linear assignment problem, so an LAP solver is the crucial component of our software. From the many existing LAP solving algorithms reviewed in section 3.2, we picked one proposed by Jonker and Volgenant [21]. The main reasons for our choice were the fact that it achieves $\mathcal{O}(n^3)$ efficiency and that a C++-implementation is readily available from [31]. Since this implementation is written in C++, we had to compile it into a dynamic linked library to use it in our source code which was written in pure C.

Another important part of the program is the computation of the cost matrix c_{ij}. For convenience, we decided to compute it from the coordinates of the water molecules' oxygen atoms only. This restriction however is somewhat arbitrary. One might as well consider building the cost matrix from the distance of the whole water molecule to its reference structure by setting

$$c_{ij} = (\vec{x}_{O,j} - \vec{x}_{0,O,i})^2 + (\vec{x}_{H1,j} - \vec{x}_{0,H1,i})^2 + (\vec{x}_{H2,j} - \vec{x}_{0,H2,i})^2.$$

Also the consideration of other criteria for the relabeling might be worthwhile. The corresponding changes should be easy to implement into the source code of g_permute.

4.1.1 Consistency checks

To test our algorithm, we applied it to a trajectory of 216 water molecules. Pictures of these results have already been presented in section 3.3. A more rigorous check for the validity was obtained by directly comparing the cost function (Eq. 3.3) for the relabeled and unrelabeled trajectory. A plot of this is shown in Fig. 4.1. Note that in this plot, the mean square displacement is shown, which is our cost function divided by the number of atoms.
While in the unrelabeled case the distance to the reference position increases

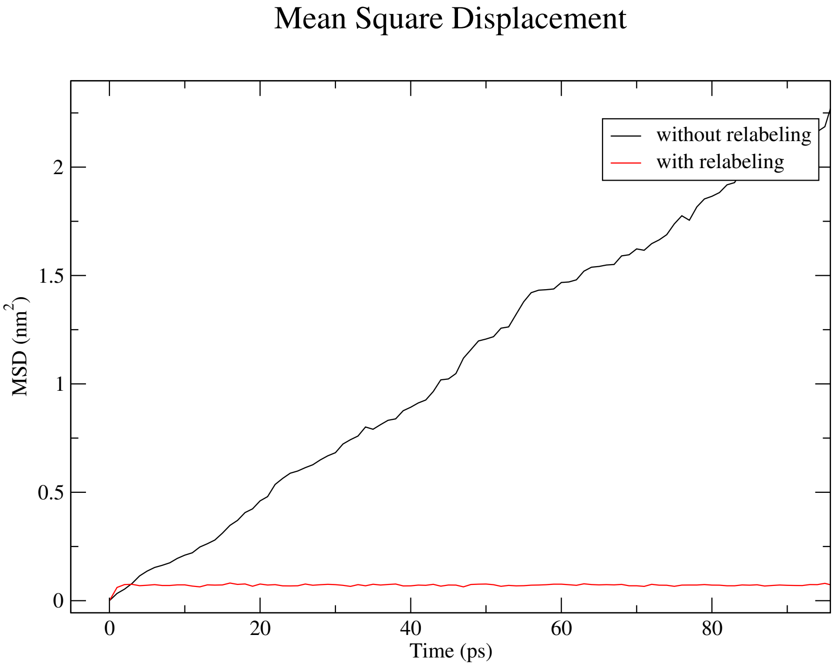

Fig. 4.1: Distance from the starting position during the simulation run. In the unrelabeled trajectory, the system moves away from its starting configuration, while in the relabeled trajectory it is constrained in its vicinity.

continously, it saturates for the relabeled case at a value of $\approx 0.07\text{nm}^2$ per atom. This value seems reasonable since it is in the range of the nearest–neighbor distance between water molecules, which agrees nicely with the similar observation of section 3.3.

The fact that in the first frames of the simulation the relabeled rmsd rises above the unrelabeled one may seem puzzling. It is however easily explained, since the relabeling minimizes only the msd between the oxygen molecules, whereas in Fig 4.1 the distance is computed using the coordinates of all water atoms. Also the fact that the tool used to compute the distance removes the effect of periodic boundary conditions adds to this effect.

4.2 Benchmark systems

The relabeling method was introduced to transform the trajectory such that its entropy can be estimated by fitting a Gaussian function to it and using the harmonic approximation. Accordingly, an important part of our work consisted in testing whether this goal is reached by applying the proposed method to various systems.

4.2.1 Molecular dynamics

To generate some of the ensembles used to test our method, we used molecular dynamics simulations. In this approach one iteratively solves Newton's equation of motion for a classical system moving in a potential $V(\vec{x})$, as was explained in section 2.1. For sufficiently long simulation times the configurations $\vec{x}(t_i)$ obtained during this simulation approximate a thermodynamic ensemble of the system. This method is capable of simulating big systems reasonably fast, so it is a natural choice for the larger ones of our test systems.

We applied a Berendsen thermostat [2] in all our MD simulations so we will assume that the generated trajectories $\vec{x}(t_i)$ approximate canonical ensembles for all simulated systems. All of our MD simulations were performed using version 3.2.1 of the program package GROMACS [30] [43] [13]. Some results were visualized using [12] and [20].

4.2.2 Monte–Carlo

For sufficiently simple systems, an ensemble can also be generated by a Monte–Carlo method. Monte–Carlo methods [38] generate ensembles by placing random "trial" configurations $\vec{x}$ in the whole configurational space and accepting only a fraction of them, depending on the potential $V(\vec{x})$, such that the accepted configurations approximate an ensemble. These methods have the advantage that a nearly ex These methods have the advantage that a nearly exact value of the entropy can be computed during the generation of the ensemble as will be explained below. We used a simple Monte–Carlo scheme to generate all those of our test systems which were simple enough to allow for this approach, whose details we present here.

Since we need only this special case, let us restrict ourselves to the case of a system moving in a "hard" potential, i.e., a potential $V(\vec{x})$ which is either 0 or ∞. Let η denote the fraction of configurational space where $V(\vec{x}) = \infty$. The Monte–Carlo method generates an ensemble by randomly choosing trial configurations $\vec{x}$ from the whole configurational space and discarding them if $V(\vec{x}) = \infty$.

To compute the entropy during the generation of the ensemble, we note that the density generated by a hard potential is zero wherever $V(\vec{x}) = \infty$. Wherever $V(\vec{x}) = 0$, the density will take on a constant value. Imposing the normalization condition, it becomes

$$\rho(\vec{x}) = \begin{cases} \frac{1}{\eta V} & \text{if } V(\vec{x}) = 0 \\ 0 & \text{if } V(\vec{x}) = \infty \end{cases}$$

The entropy thus becomes

$$S_{\mathrm{MC}} = -k \int_{\eta V} d^6x \frac{\ln \frac{1}{\eta V}}{\eta V} = k\left(\ln \eta + \ln V\right)$$

The quantity η can be computed during the generation of the ensemble. Since the trial configurations are chosen with uniform probability from the whole configurational space, η is given by the fraction of configurations rejected by the Monte–Carlo method.

4.2.3 Van–der–Waals gas

As a first test case we applied our method to a gas of Lennard–Jones particles. This system is simpler than real solvents but should nevertheless exhibit typical solvent effects.

We simulated an ensemble of 100 atoms of mass 39.948 amu each interacting only via a van–der–Waals interaction stemming from the Lennard–Jones–potential

$$V_{LJ}(r_{ij}) = \frac{C_{12}}{r_{ij}^{12}} - \frac{C_6}{r_{ij}^{6}} \qquad \text{with } r_{ij} = |\vec{x}_i - \vec{x}_j| \tag{4.1}$$

characterized by the coupling constants C_{12} and C_6.
The simulation was performed using molecular dynamics. The system was simulated for 50 ns at a temperature of 300K. The starting configuration was obtained from an energy minimization. We did not apply a barostat since constant volume is a natural choice for a gas. Instead, the volume of the simulation box was held constant at $(2.2\text{ nm})^3$. This is equivalent to a pressure of 385 bar. This high pressure allows us to sample a wide range of the interaction strength without the disturbing effects of phase transitions. We used periodic boundary conditions for the simulation box.

The system was simulated for 19 values of the coupling constants ranging from $(C_6, C_{12}) = (0, 0)$ (ideal gas) up to $10 \times (1.2340 \cdot 10^{-3}, 2.3574 \cdot 10^{-6}\text{nm}^6)\text{kJnm}^6\text{mol}^{-1}$. The parameters $(1.2340 \cdot 10^{-3}, 2.3574 \cdot 10^{-6}\text{nm}^6)\text{kJnm}^6\text{mol}^{-1}$ will serve as a reference in further plots and calculations. Instead of exact numerical values we will parameterize the coupling constants by an interaction parameter λ such that

$$(C_6, C_{12})(\lambda) = \lambda \times (1.2340 \cdot 10^{-3}, 2.3574 \cdot 10^{-6}\text{nm}^6)\text{kJnm}^6\text{mol}^{-1}.$$

The parameters of some real gases can be found in [16]. For the rare gas Argon they are $(6.21 \cdot 10^{-3}, 9.68 \cdot 10^{-6}\text{nm}^6)\text{kJnm}^6\text{mol}^{-1}$, so our model gas, taken at $\lambda = 5$, is roughly equivalent to Argon.

To simulate the solvation of a particle in a solvent, we performed a second series of simulations where the interaction strength of the particles was held constant at $\lambda = 1$ and an additional particle was inserted at the center of the simulation box. The position of this particle was restrained and its interaction was varied between $\lambda = 0$ and $\lambda = 5000$.

All resulting trajectories were relabeled using g_permute. We used different reference structures: The first frame of the trajectory and an artificial reference structure in which the atoms were lined up along a simple cubic (sc) lattice. For the simulation series with the additional inserted particle, we also used the last frame as a reference structure. Covariance analyses were carried out for all resulting trajectories, using the GROMACS toolbox.

4.2.4 3x2–dimensional model

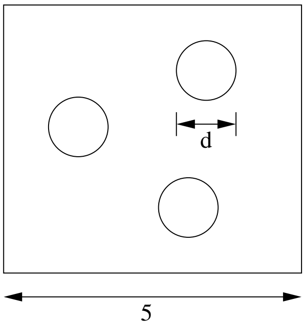

Fig. 4.2: 3x2–dimensional model system

To understand the behavior of the algorithms in the test systems discussed above, we analyzed an even simpler system, namely three disks of diameter d moving an a 2D box. This system is sufficiently simple to generate the ensemble by the Monte–Carlo method explained above.

The interaction potential of two 2–dimensional disks placed at positions $\vec{x}_1$ and $\vec{x}_2$ is

$$V(\vec{x}) = \begin{cases} \infty & \text{if } |\vec{x}_1 - \vec{x}_2|^2 < d^2 \\ 0 & \text{otherwise} \end{cases}$$

The trial configurations were randomly placed on a square of sidelength 5. We generated several ensembles with the diameter of the disks ranging from $d = 0$ to $d = 3.0$ in steps of $\Delta d = 0.2$. All ensembles obtained in this way were relabeled using our standard LAP solver [21] using the three points $[3.8, 0.6], [1.2, 2.9]$ and $[4.4, 2.7]$ as reference structure.

4.2.5 Water

Since we aim at a treatment of more complex solvents such as water, we also tested our algorithm with a simple water system. We simulated 216 water molecules for 50ns in a box of approximately 1.87nm sidelength with periodic boundaries. Again we employed molecular dynamics and solved Newton's equation of motion to generate a trajectory of this system. We used the simple point charge (SPC) model [3] which describes the water molecule by three atoms (O, H1 and H2) held together by rigid sticks. The bond lengths were fixed using the SETTLE algorithm [34]. The partial charges of the atoms were taken from the GROMACS force field [30] [4].
The system was coupled to a 300K temperature bath and a 1 bar pressure bath using a Berendsen thermo- and barostat [2], since an isobaric and isothermal

system is the typical choice for water. The resulting trajectory was relabeled using g_permute and using the first frame as reference. Covariance analyses of the relabeled trajectory were carried out including atoms of the system as well as, alternatively, its oxygens atoms only.

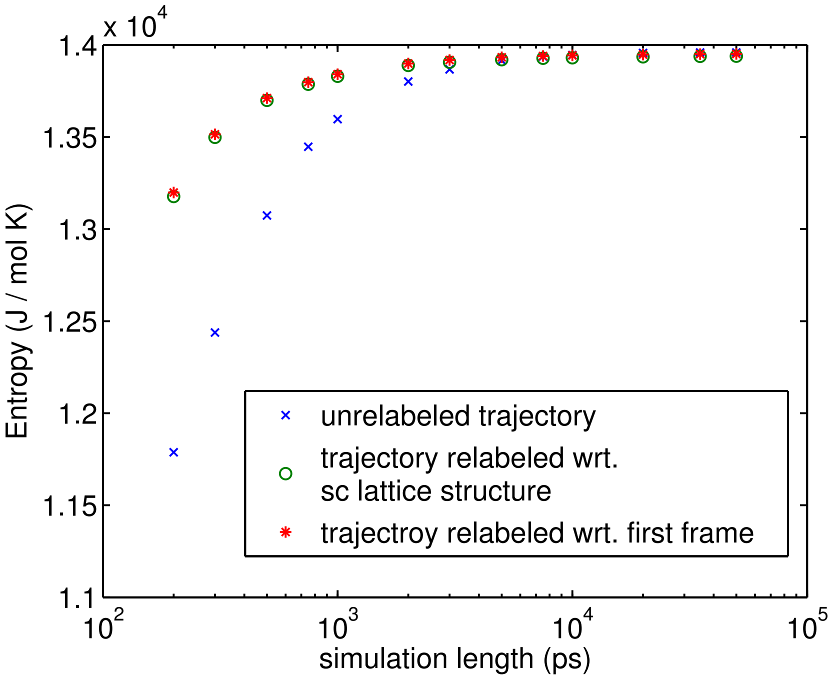

Fig. 5.1: Convergence of the simulation. As expected, the entropy estimate of the relabeled trajectories converges faster than for the unrelabeled trajectory. All simulations converge within the 50 ns simulation time.

5 Results and Discussion

5.1 Entropy of a van–der–Waals gas

Let us now analyze the entropy estimation provided by our method for the first test system, the van–der–Waals gas.

5.1.1 Convergence analysis

Since convergence of the simulation is a critical issue [18] [19] [17], we first performed a convergence test of the simulation. To this aim, we considered the ideal gas case and computed the entropy of the unrelabeled and relabeled trajectory using the Schlitter formula. In the relabeled case the entropy was corrected using Eq. 3.6. Fig. 5.1 show the results for several simulation lengths.

As can be seen, the entropy converges to a final value within less than 10^4 ps. Hence, a simulation length of 50 ns should be sufficient to suppress sampling errors. The generated trajectories can thus be used to compare different entropy estimation methods. It is also apparent from the figure that the relabeled trajec-

tory converges significantly faster than the unrelabeled one. This result is exactly what we expected, since the relabeling maps the trajectory into a small volume of the phase space, which is in turn sampled faster.
Let us now turn to the analysis of the entropies for different values of λ.

5.1.2 Ideal gas

Let us consider the case $\lambda = 0$ first. Here, the entropy of the system as well as the estimate given by Schlitter's formula can be calculated analytically as described in section 2.5.1.

There we found that the entropy of an ideal gas under the given conditions is $S = 13.54\frac{\text{kJ}}{\text{K mol}}$ and that estimating its entropy using the Schlitter approach should lead to a value of $S = 13.98\frac{\text{kJ}}{\text{K mol}}$

The entropy of the unrelabeled trajectory was computed using the Schlitter method, which yields an entropy of $S_{\text{unrelabeled}} = 13.95\frac{\text{kJ}}{\text{K mol}}$ which agrees nicely with the calculations presented above.

Computing the entropy from the relabeled trajectory using the Schlitter formula and correcting the result by formula 3.6 yields the entropies

$$S_{\text{perm, sc}} = 13.94\frac{\text{kJ}}{\text{K mol}}, \text{ and } S_{\text{perm, frame}} = 13.95\frac{\text{kJ}}{\text{K mol}}$$

depending on which reference structure (sc lattice or first frame) was used for the relabeling. The good agreement with section 2.5.1 is encouraging. It shows that the slice of the phase space which remains accessible to the relabeled trajectory can be well fit by a Gaussian function. This is a nontrivial result, since the properties derived in section 3.4 provided no hints to whether a Gaussian function is well adapted to the shape of the slice.

5.1.3 Excess Entropies – the case $\lambda \neq 0$

Let us now turn to the case $\lambda \neq 0$. Since we have already discussed the magnitude of the absolute entropy in the previous section, we now switch to the consideration of relative entropies. Fig. 5.2 shows the **excess entropy**, i.e., the entropy difference to the ideal gas. We compute reference values of the entropy at different λ by performing a thermodynamic integration (TI) between the ideal gas and the gas at a finite λ.

The growing interaction leads to a decrease of the entropy, as is shown by the decreasing TI curve. The estimate based on the PCA of the unrelabeled trajectory fails to recognize this effect. it even increases slightly for increasing λ.

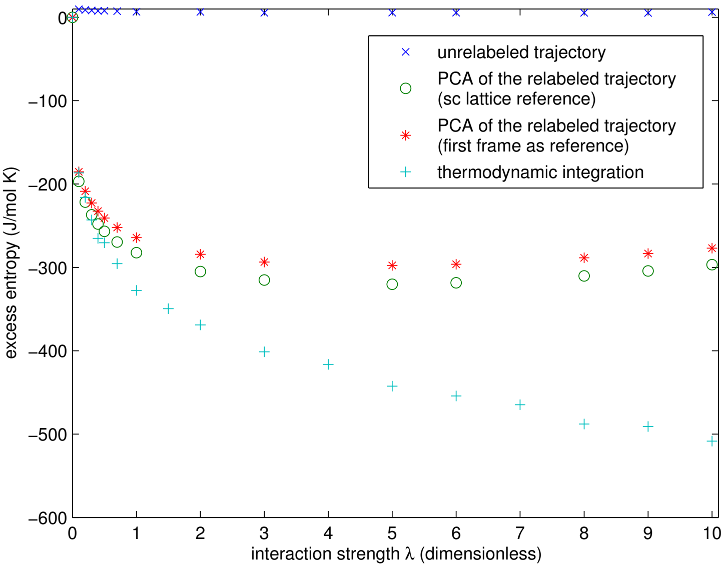

Fig. 5.2: Excess entropies of a van–der–Waals gas as a function of the interaction strength. Thermodynamic integration (cyan curve) is used as a benchmark for the entropy estimate provided by a PCA of the unrelabeled and the relabeled trajectories.

The accuracy of the entropy estimate is dramatically enhanced by the relabeling procedure. Both estimates based on relabeled trajectories clearly recognize the decrease of the entropy. However they do not fully reproduce the TI result which becomes particularly clear for the case of large λ. The dependence on the reference structure used is small. Since the reason for the observed properties is not obvious, let us analyze the 3x2D model to explain the results of Fig. 5.2.

5.2 3x2–dimensional models

In this case, the Monte–Carlo algorithm provides a nearly exact value for the entropy, which we will use as a benchmark for the PCA–based results. The configurational entropy was computed for every d using a PCA of the unrelabeled as well as the relabeled trajectory. Since we are only interested in configurational entropies in this case, the Karplus formula (Eq. 2.14) is used for the computation. The resulting entropies are shown in Fig 5.3 The agreement of the ideal–gas entropies between the PCA estimates and the “exact” result (Monte–Carlo) is not as good as in the case of the van–der–Waals gas. This is most likely due to the different box size.

For varying d the situation is similar to the van–der–Waals example: The Monte–Carlo entropy decreases for increasing d. A PCA of the unrelabeled ensemble fails

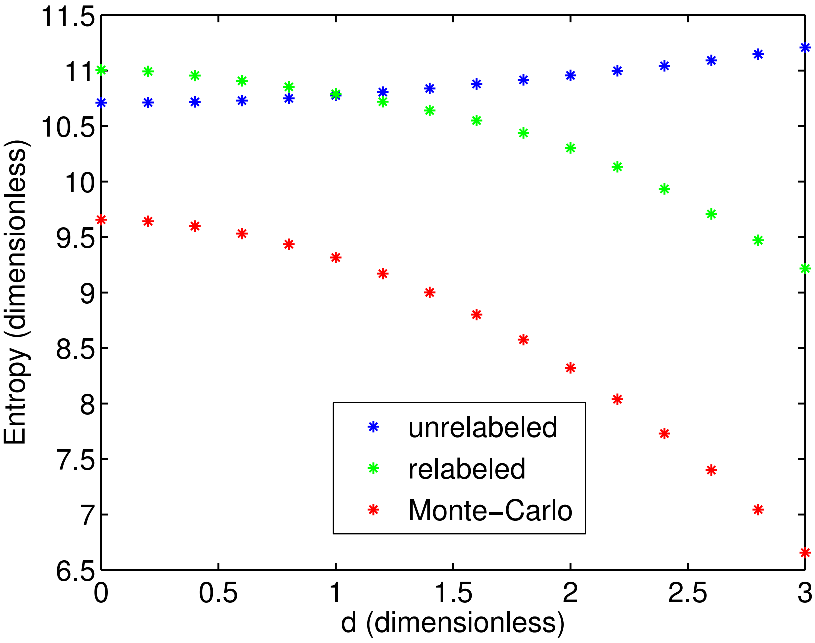

Fig. 5.3: Entropies in the 3x2D case. Here too, the estimate based on the relabeled ensemble yields qualitatively right results, in contrast to the estimate based on the unrelabeled ensemble.

to recognize this trend. If, however, the relabeled ensemble is used to compute the PCA, the entropy estimate is clearly enhanced. It reproduces the right trend, namely a decreasing entropy for increasing d. Its entropy estimate is, however not fully exact which can be seen best by direct comparison of the excess entropies given by Monte–Carlo and PCA (Fig 5.4). To find the reason for the different behavior of relabeled and unrelabeled trajectory, we analyzed the eigenvalues of the covariance matrix of the unrelabeled trajectory, since these values determine the entropy estimate of a PCA by means of Eq. 2.14. The spectrum for several values of d is shown in Fig 5.5

The spectrum behaves as predicted in section 2.5.2. It is split into two groups of $d = 2$ and $(N-1)d = 4$ eigenvalues each. They correspond to the center–of–mass eigenvectors and the relative eigenvectors respectively.

For increasing r, the eigenvalues of the center–of–mass eigenvectors decrease while the eigenvalues of the relative eigenvectors increase. This suggests that the increase in entropy in Fig 5.3 is due to changes of the density in the relative subspace. A scatter plot of the unrelabeled and relabeled density is shown in Fig 5.6.

At $r = 0$ the ensemble would fill the whole subspace ranging from -5 to 5 in the

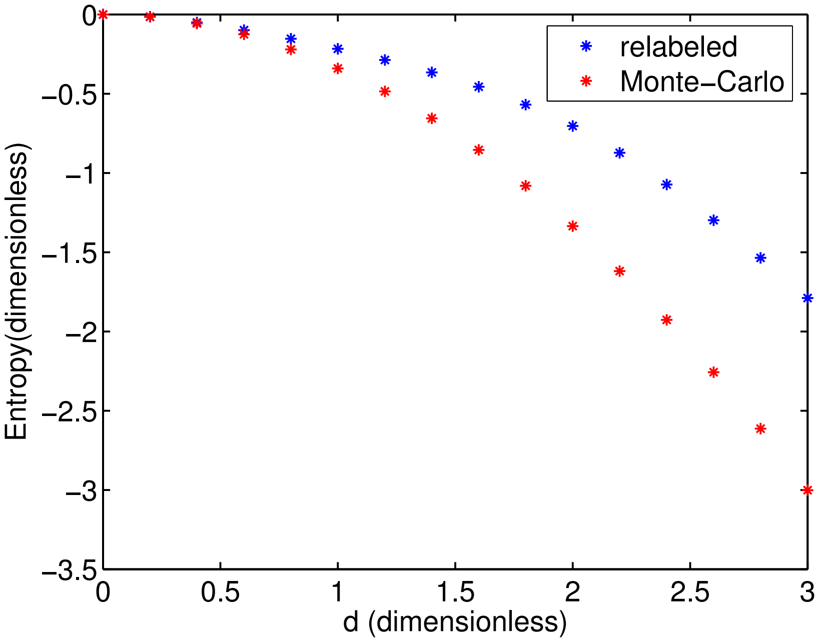

Fig. 5.4: Excess Entropies in the 3x2D case as estimated by MC and a PCA of the relabeled trajectory . The PCA recognizes the right trend, its results however do not exactly reproduce the Monte–Carlo entropy.

x and y–direction. Introducing a finite diameter of the disks cuts a spherical hole into the density around the origin, since two disks overlap for small $\vec{x}_2 - \vec{x}_1$.

The relabeling procedure constrains the density in an intersection of half-spaces. The clear cut visible in Fig 5.6 stems from the permutation 123 → 213, since this forces the ensemble to lie in the half-space

$$\vec{x}_0^T(1 - P_{213})\vec{x} > 0$$

Its normal vector $\vec{x}_0^T(1 - P_{213})$ lies in the $\vec{x}_1 - \vec{x}_2$ subspace which explains the clearly visible cut. The effect of the other permutations can be seen from the fact that the relabeled density vanishes also in the upper right and lower left corner.

Since the growing disk diameter excludes a volume in the center of this subspace, the variances of a PCA will increase for increasing r, which explains the growing eigenvalues in the relative subspaces in Fig 5.5. This also explains why the increase in entropy for increasing interaction parameters in the PCA of the unrelabeled ensemble is more pronounced in the 3x2D example than for the van–der–Waals gas. In the latter case, the attractive r^{-6} contribution to the potential energy decreases the variance for increasing λ.

The situation is different for a PCA of the relabeled system. The relabeling

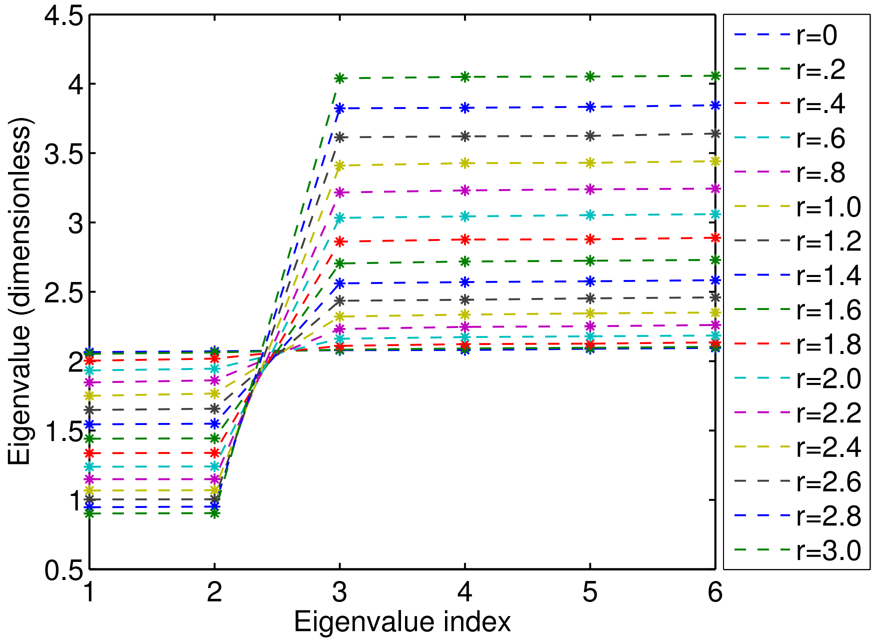

Fig. 5.5: PCA eigenvalues of the unrelabeled trajectory for varying d. As predicted in section 2.5.2 the 2 eigenvalues corresponding to the center–of–mass eigenvectors can be clearly distinguished from the 4 eigenvalues of the relative eigenvectors.

algorithm reduces the phase space to a slice delimited by planes running through the center of the relative subspaces. The "holes" produced by the repulsion of the disks will therefore be located at the boundary of the slice. A Gauss fit of the relabeled density is shown in Fig. 5.7.

It is apparent from this figure that, in the case of a relabeled ensemble, a growing hole will reduce the variances of the PCA in the relative subspace. This of course does not mean that the PCA eigenvalues will also decrease for a PCA of the complete configurational space, since in this case the relative subspaces are not necessarily eigenspaces of the PCA any more. However since a decrease of the entropy is in fact observed for the relabeled trajectory and increasing r, we suggest that it can be attributed to this effect.

It is also apparent from Fig 5.7 why the result will deviate from the exact result. The relabeled ensemble is not at all distributed according to a Gaussian distribution.

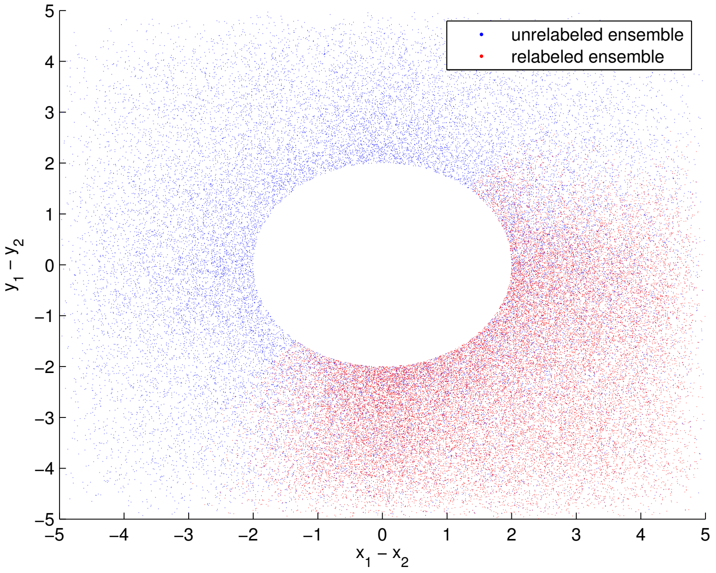

Fig. 5.6: Densities in the $\vec{x}_2 - \vec{x}_1$ subspace for the $d = 2.0$ case. Configurations near the origin are forbidden, since the disks would overlap here. The relabeling constrains the density in an intersection of half-spaces, whose border is visible here as the line running from the lower left to the upper right which delimits the relabeled ensemble. ensemble.

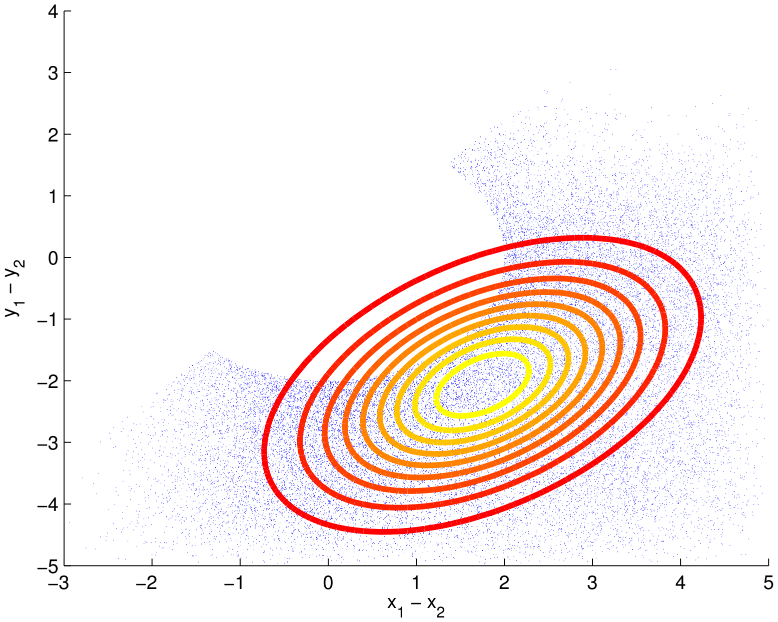

Fig. 5.7: Gauss fit of the relabeled ensemble. The rings are the contours of the Gaussian density distribution estimated by the quasi-harmonic approximation.

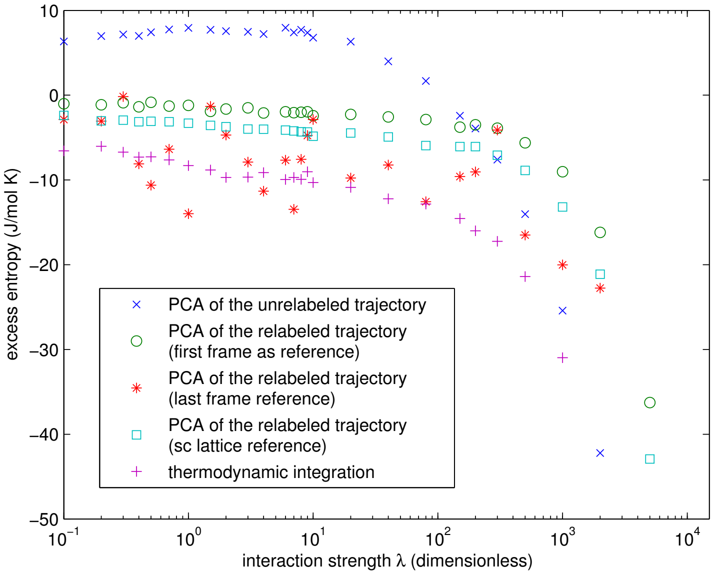

Fig. 5.8: Entropy estimate for the solvation series. The relabeling with respect to the starting structure or the lattice structure leads to a significant improvement of the entropy estimate.

5.3 Solvation entropies

The entropy of a pure solvent, which we discussed in the previous sections, is an important benchmark for our method. It is however a rather unrealistic test system. In practice one is more often interested in the solvation entropies of particles within the solvent and not in the entropy change for different values of the solvent interaction. Let us therefore analyze the entropy estimates for the simulation series in which we inserted the additional particle at the center of the simulation box. This series will be referred to as "solvation series" below.

As in the case of the pure van–der–Waals gas, the entropy was computed by a PCA of the unrelabeled as well as the relabeled trajectories. Thermodynamic Integration was used as a reference. The results of these simulations are shown in Fig. 5.8.

For high values, the attractive r^{-6} contribution to the interaction becomes so strong that the particles begin to condense around the central particle. This effect is even recognized by a PCA of the unrelabeled trajectory, since it decreases the variance of the particles.

In the case of a weakly interacting central particle (λ small), the situation is similar to the pure van–der–Waals gas: When the interaction of the central particle is switched on (the step from $\lambda = 0$ to $\lambda = 0.1$), its repulsive core pushes the

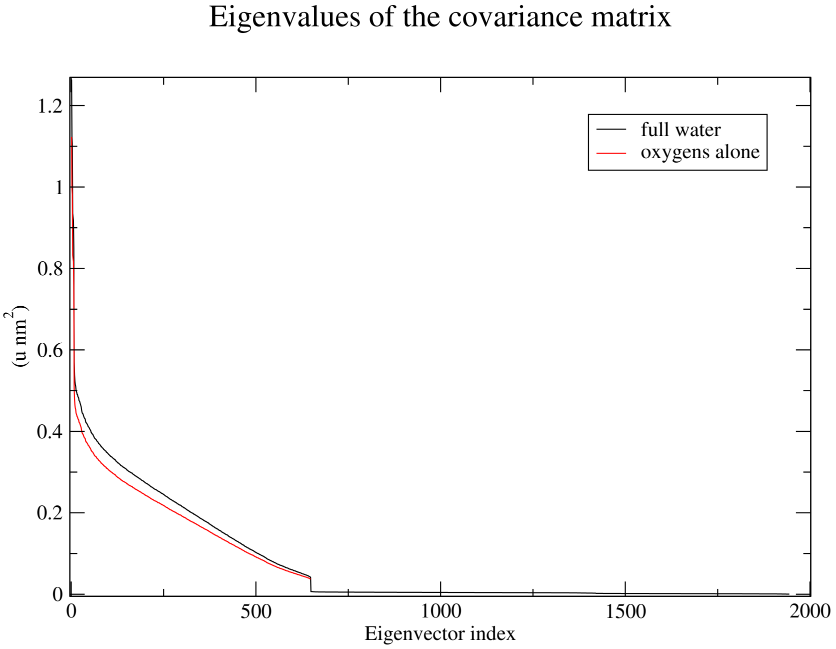

Fig. 5.9: Eigenvalue spectrum of a PCA of water. The black curve is obtained from the covariance matrix of all atoms. The red curve is obtained from the oxygens alone. The small difference vanishes nearly completely when the oxygens' masses are rescaled to the mass of a whole water molecule.

particles away from the center, which is wrongly recognized as an increase in entropy by the PCA of the unrelabeled trajectory.
As in the previous study, the entropy estimate is significantly enhanced when the relabeled trajectory is used as basis for the PCA. In this case, the decrease of the entropy for increasing λ is predicted right. The transition to the condensed regime, which sets in at $\lambda \approx 10^2$ is recognized more precisely than by the PCA of the unrelabeled trajectory.

The entropy estimate of the relabeled trajectories depends on the reference structure used. When the lattice or the first frame is used, the entropy change is predicted qualitatively right, using the last frame as reference for the relabeling introduces a significant amount of error. This, however, does not come as a surprise since this structure varies significantly between the simulations.

5.4 Entropy of water

Finally let us examine the water system. Here we computed the covariance matrix twice, once including all atoms of the box and once including only the oxygen atoms. The resulting eigenvalue spectra are shown in Fig 5.9:

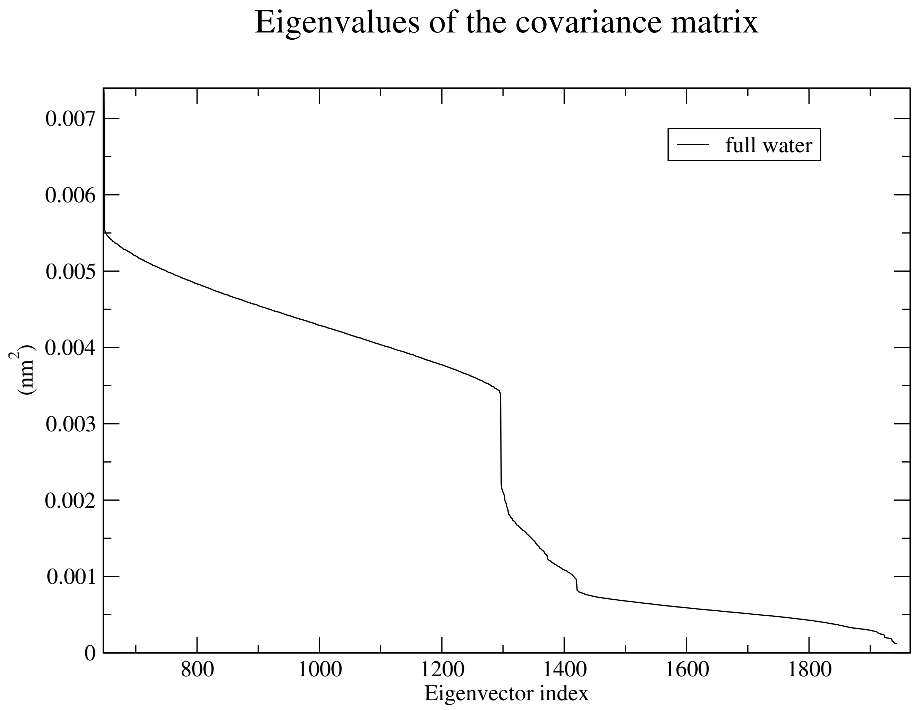

Fig. 5.10: Close–up on the last 1296 eigenvalues of the covariance matrix of the full water molecules.

Let us discuss the first 648 eigenvalues first. They show hardly any difference between the two analyses. The small difference visible vanishes nearly completely when the oxygens' masses are rescaled to the mass of a full water molecule. The analysis restricted to the oxygens can only reveal the translational contribution to the movement of the water molecules. The fact that this analysis yields nearly the same values for the first first 648 eigenvalues suggests that the PCA of the relabeled trajectory also distinguishes between translational and rotational movement and that its first 648 eigenvalues belong to the translational movement of the whole water molecules.

The information about the rotational motion of the molecules must therefore be encoded in the 1296 additional eigenvalues provided by the covariance analysis of the full water molecules. These eigenvalues show up as the tail of small eigenvectors with index > 648 in the "full water" curve of Fig. 5.9.

A close–up plot of these eigenvalues is shown in Fig. 5.10. Note that in this plot, the covariance matrix is built using non mass–weighed coordinates.

These eigenvalues can again be split into two groups. The eigenvalues of the first group have values of approximately 0.004 nm^2. The second group takes values in the 0.001nm^2 range. The movements giving rise to these values must thus lie in the range of .06nm and .03nm respectively. Intramolecular distances lie in this range which suggests that the last 1296 eigenvectors are intramolecular vectors.

This conclusion is confirmed by an analysis of the eigenvectors, which we performed as follows. We divided each eigenvector $\vec{x}$ into the 3N 3D–vectors $\vec{x}^i_O, \vec{x}^i_{H1}, \vec{x}^i_{H2}$, $i \in 1, \dots N$ of the oxygen and hydrogen coordinates. The 9D–vector of one water molecule will be denoted by $\vec{x}_i$. We computed the projection of the full water vectors $\vec{x}_i$ onto the subspaces of center–of–mass motion and intramolecular motions. Their norms averaged over all molecules are given by

$$\begin{aligned}
\langle \mathrm{com} \rangle &= \frac{1}{N} \sum_{i=1}^{N} \frac{\left\| \frac{1}{\sqrt{3}} (\vec{x}^i_O + \vec{x}^i_{H1} + \vec{x}^i_{H2}) \right\|}{\|\vec{x}_i\|} \\
\langle \mathrm{oh} \rangle &= \frac{1}{N} \sum_{i=1}^{N} \frac{\left\| \frac{1}{\sqrt{2}} (\vec{x}^i_{H1} - \vec{x}^i_O) \right\|}{\|\vec{x}_i\|} \\
\langle \mathrm{hh} \rangle &= \frac{1}{N} \sum_{i=1}^{N} \frac{\left\| \frac{1}{\sqrt{2}} (\vec{x}^i_{H2} - \vec{x}^i_{H1}) \right\|}{\|\vec{x}_i\|}
\end{aligned}$$

We will use these quantities as a measure of how much an eigenvector is composed of translational motion and intramolecular motions. The values of these numbers for all eigenvectors is shown in Fig 5.11

Clearly, the first 648 eigenvectors are composed nearly exclusively of translational contributions, whereas the last 1296 vectors consist of intramolecular vectors. Vectors 649 to 1296 consist mainly of H-H vectors while the last 648 vectors correspond to the O-H vector. This is also reflected in the last 648 eigenvalues being smaller than the eigenvalues in the first two thirds of the spectrum.

The fact that eigenvectors have significant contributions from intramolecular vectors may seem puzzling since in the SPC water model, the OH–bonds are fixed and the atoms inside one molecule cannot move with respect to each other. This effect is explained by the following consideration: During the simulation, the hydrogen atoms of the water molecules rotate around their respective oxygen atom. When computing the average configuration, this rotation is averaged over, so the hydrogen atoms will occupy the same position as their respective oxygen atoms. The O-H–vector will thus have a nonzero variance, even though its mean value vanishes, since the hydrogen is in every ensemble element separated from the oxygen. Accordingly, eigenvectors of the PCA can be composed of intramolecular vectors, even if the bond lengths are held fixed. Computing the entropy from this eigenvalue spectrum by means of the quasi-harmonic approximation one would assume motions along these vectors, which is clearly unphysical. One might argue that due to their small values the intramolecular eigenvalues are discarded

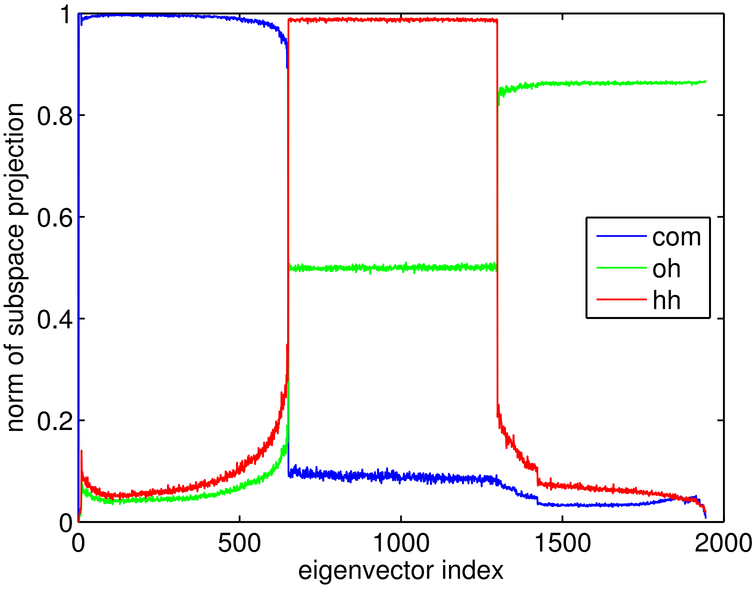

Fig. 5.11: Norms of the projections of the eigenvectors into the center–of–mass (com), the O-H difference (oh) and the H-H difference subspace (hh). The first 648 eigenvectors are composed mainly of translational contributions, whereas intramolecular vectors dominate in the last 1296 vectors.

anyway by the Schlitter formula. Since however these eigenvalues contain all the information about angular correlations of the water, one also discards important contributions to the entropy in that way. (see [28] [32] for a discussion of these contributions)

These results show that the applying a PCA to a relabeled trajectory of water one automatically obtains a decomposition into translational and rotational contributions to the motion. The Karplus or Schlitter approach should be applicable to the translational contribution similarly well as to the van–der–Waals gas, since the translational movement of the water molecules is similar to a gas of interacting particles without rotational degrees of freedom.

The situation is more difficult for the rotational contributions: Here, a PCA of the relabeled trajectory in Cartesian coordinates is not suited to extract information about the rotation of the water molecules. One obvious approach would be to introduce angular coordinates for the water orientation. In this way one could prevent the intramolecular vectors from consuming two important thirds of the eigenvalue spectrum. However we did not pursue this approach further and rather focused on more general improvements of our method, which will be explained in chapter 6.

5.5 Discussion

The tests for the van–der–Waals gas and the 3x2D model yield a consistent picture: The relabeling procedure leads to an improvement of the entropy estimate and can be used to observe effects which would not be seen in a PCA of the unrelabeled trajectory. An example of this is the entropy decrease for increasing interactions, which is observed both in the van–der–Waals and the 3x2D model. Here, a PCA of the unrelabeled trajectory computes an increasing variance and entropy, whereas a PCA of the relabeled trajectory clearly recognizes the right trend. The solvation simulations yield a similar picture: The relabeling algorithm improves the density estimate and reveals a far more detailed picture of the entropy change with growing interaction than a PCA of the unrelabeled ensemble. Also the PCA of the relabeled trajectory, however, does not exactly reproduce the nearly exact result of the TI or Monte–Carlo estimates.

Returning to the two big obstacles mentioned in the introduction – bad sampling and the difficulty to find a fit function for the solvent density – we have shown that at least the first problem is significantly alleviated by our approach: The entropy estimate converges far faster for the relabeled trajectory than for the unrelabeled one.
We also contributed to the solution of the second problem by showing that the relabeling procedure restricts the ensemble to a part of the phase space in which the topology of the density is sufficiently simple for analytical density estimates to become applicable. The harmonic approximation, however, is not the best choice to produce accurate entropy estimates of the relabeled trajectory as can be seen from the deviation of this estimate from the TI or Monte–Carlo results. We have identified the reason for this by showing in our 3x2D simulations that the relabeled density – though it is more localized than the original one – is still too anharmonic to be fit by a Gaussian function. (Fig. 5.7)

The analysis of the water trajectory reveals that a PCA of the relabeled trajectory automatically distinguishes between translational and rotational parts of the motion. The latter part presents a challenge which has to be tackled in order to apply our approach to biologically relevant systems: Computing a covariance matrix in Cartesian coordinates is not sufficient to treat angular degrees of freedom.

6 Improvements

The insights of the previous section suggest several ways to improve the density estimate. None of them has evolved into a full–scale implementation yet, however we can present proof–of–principle calculations for some of them, which are presented in this chapter.

First we investigated whether one might be able to get better estimates by purely analytical means such as switching to another coordinate system.
An intriguing possibility to refine the density estimation is the connection of our method with thermodynamic integration. Since the difference between our entropy estimation and the "true" entropy is small, we examined whether TI can be used to bridge this gap.
One may also think about transforming the density itself instead of finding a better fit function. Experiments in this direction are presented in section 6.4.

6.1 Analytical improvements

6.1.1 Use of Monte–Carlo integration in the Karplus approach

Both the Karplus and the Schlitter approach consist of fitting an analytical ansatz for ρ to the trajectory and evaluating

$$S = -k_B \int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\, \rho \ln(\rho)$$

analytically. At least in the latter part one may spot space for improvement. It is not necessary to integrate analytically over the (possibly inaccurate) density estimation ρ. By remembering the fact that the trajectory approximates an ensemble one may replace the integral over the density by the time average as follows

$$\int_\Gamma d^{3N}\vec{x}\, d^{3N}\vec{p}\, \rho(\vec{x},\vec{p}) \ln(\rho(\vec{x},\vec{p})) = \frac{1}{T}\int dt \ln\left[\rho(\vec{x}(t),\vec{p}(t))\right] \tag{6.1}$$

where T is the simulation length and t the time parameter of the simulation. A density estimation is however still required since the term $\ln(\rho)$ cannot be expressed by a time average.

It can be shown that in the case of a Gaussian ansatz for the density this method leads to no improvement. To see this, one has to insert the Gaussian ansatz

$$\rho(\vec{x}) = \frac{e^{-\frac{1}{2}\vec{x}^T \mathbf{C^{-1}} \vec{x}}}{Z}$$

into Eq. 6.1.

$$\begin{aligned} S &= k_B\frac{1}{T}\int dt\left[\ln Z+\frac{1}{2}\vec{x}^T\mathbf{C}^{-1}\vec{x}\right]=k_B\ln Z+\frac{k_B}{2T}\sum_{ij}\int dt(\mathbf{C}^{-1})_{ij}\,x_ix_j= \\ &= k_B\ln Z+\frac{k}{2}\mathrm{tr}(\mathbf{C}^{-1}\mathbf{C})=k_B\left[\frac{3N}{2}+\ln Z\right] \end{aligned}$$

which is exactly the Karplus result (Eq. 2.14). Note however that this is true only for a Gaussian ansatz. For different fit functions this method may very well lead to an improvement.

6.1.2 Transformation to other coordinate systems

In chapter 5.2, the accuracy of the entropy estimation from a relabeled trajectory has been shown to suffer from anharmonicity of the free energy landscape. The regions of highest anharmonicity have been found to lie in the relative subspaces and to be spherically symmetric around the origin. This result suggests to introduce spherical coordinates in the relative subspaces to improve the entropy estimation.

To test this approach we again chose the 3x2D system. We transformed the coordinates of the relabeled ensemble into the basis of center–of–mass and two relative coordinates ($\vec{x}_1+\vec{x}_2+\vec{x}_3$, $\vec{x}_2-\vec{x}_1$ and $\vec{x}_3-\vec{x}_2$ respectively). Let B denote the transformation matrix of this base change. The relative coordinates were transformed into polar coordinates (r_1,ϕ_1) and (r_2,ϕ_2) as suggested above.

The density was estimated from the transformed ensemble by means of a PCA in the new coordinates. From this density estimate ρ_{pol} one has to recover the density in the old coordinates, since this density determines the entropy by means of Eq. 2.4. We used two different approaches to solve this problem:
In the first approach, we made use of the results of the last section and replaced the phase space integration by the ensemble average (Eq. 6.1). The density in the logarithm was computed from ρ_{pol} by correcting it for the volume change induced by the various transformations, i.e, by rescaling it by a factor $(\det(B)\cdot r_1\cdot r_2)^{-1}$. Since this scale factor depends on the position in phase space, the density estimation will deviate from a Gaussian distribution when it is transformed back into the old coordinates. The entropy estimation obtained in this way is referred to in Fig. 6.1 as "PCA in polar coordinates (MC integration)"
In the second approach, we modified the transformation such that it does not induce a volume change. To this aim, we transformed the radial coordinates r_1 and r_2 into r_1', r_2' with $r_1'=r_1^2/2$ and $r_2'=r_2^2/2$. In this way, the volume element in the new coordinates becomes

$$dr_1'd\phi_1=r_1dr_1d\phi_1$$

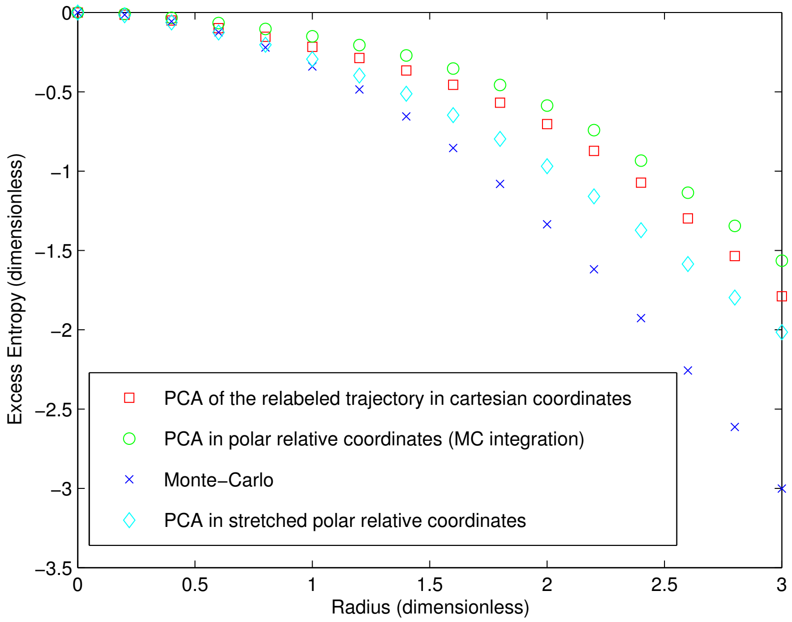

Fig. 6.1: Excess entropies in the 3x2D case. The entropy estimation can be improved by the use of polar coordinates (green and cyan curve), although it still differs from the true entropy (dark blue curve).

which is exactly the volume element in the unstretched polar coordinates r_1, ϕ_1. Furthermore, we rescaled B such that $\det(B) = 1$. Then the whole transformation is volume–conserving and the entropy can be computed directly by using the Karplus formula for a PCA of the transformed ensemble. The entropy estimation obtained in this way is referred to in Fig. 6.1 as "PCA in stretched polar relative coordinates"

In the case of stretched polar coordinates the estimation becomes slightly better, it remains however far from accurate. This is not surprising, since a transformation into radial coordinates will not suffice to transform the density into Gaussian shape. Even stronger problems arise when this scheme is applied to higher particle numbers. Here the two–particle–interactions will give rise to N^2 "holes", but only N of them can be treated by the introduction of radial coordinates.

One might think about different coordinate systems. However we are not aware of a coordinate system adapted to the symmetry of N particles interacting by pair interactions.

6.2 PCA–TI hybrid approach

The entropy estimate obtained by applying Schlitter's formula to a PCA of the relabeled trajectory does not yield quantitatively correct results. The difference to the real entropy however is small as was discussed in section 5.1. Since thermodynamic integration is capable of computing small entropy differences, we suggest to use the Karplus or Schlitter estimation of the entropy of the relabeled ensemble as a basis and compute its deviation from the real entropy by means of TI (see Fig. 6.2 for an illustration). This method was originally proposed for the case of free energy by Stoessel and Nowak [41].

TI computes entropy differences by gradually changing the Hamiltonian during the simulation. Since the Karplus and Schlitter methods assume the Hamiltonian of the system to be harmonic, one has to perform a TI between the Hamiltonians

$$H_{\text{ha}} = \sum_{i=1}^{N} \frac{\vec{p}_i^2}{2m_i} - k_B T (\vec{x} - \vec{x}_0)^T \boldsymbol{C}^{-1} (\vec{x} - \vec{x}_0) \qquad \text{and} \qquad H_{\text{sys}} = \sum_{i=1}^{N} \frac{\vec{p}_i^2}{2m_i} - V(\vec{x})$$

The fact that the density estimation in the Karplus and Schlitter approach stems from a Hamiltonian is crucial here.

6.2.1 Subspace entropies and the PCA–TI hybrid approach

The above approach assumes that the PCA is run over the coordinates of all molecules in the system, i.e, over protein as well as solvent coordinates. Often however one is also interested in the entropy of the protein subspace and PCA combined with the Karplus/Schlitter technique is the usual choice to compute it. We therefore investigated whether the procedure of section 6.2 can also be applied to refine an estimate of a subspace entropy .

A quasi-harmonic Hamiltonian restricted to the subspace can easily be derived. The crucial step for this approach to work is thus to find a variant of thermodynamic integration, which computes the change of entropy in a subspace when the Hamiltonian is changed.
Such a formula however cannot be derived as straightforward as the original TI formula (Eq. 2.19). To see this, let the derivative $d/d\lambda$ be denoted by a prime ($x' = dx/d\lambda$). Also, we will for the sake of simplicity omit the position arguments of all Hamiltonians in this section. Let furthermore $H = H_S + H_I + H_P$ be the Hamiltonian of the system composed of solvent, interaction and protein contributions respectively.
Recalling the definition 2.5

$$\rho_p = \frac{e^{-\beta H_p} \int e^{-\beta H_I} e^{-\beta H_s} dx_s}{Z}$$

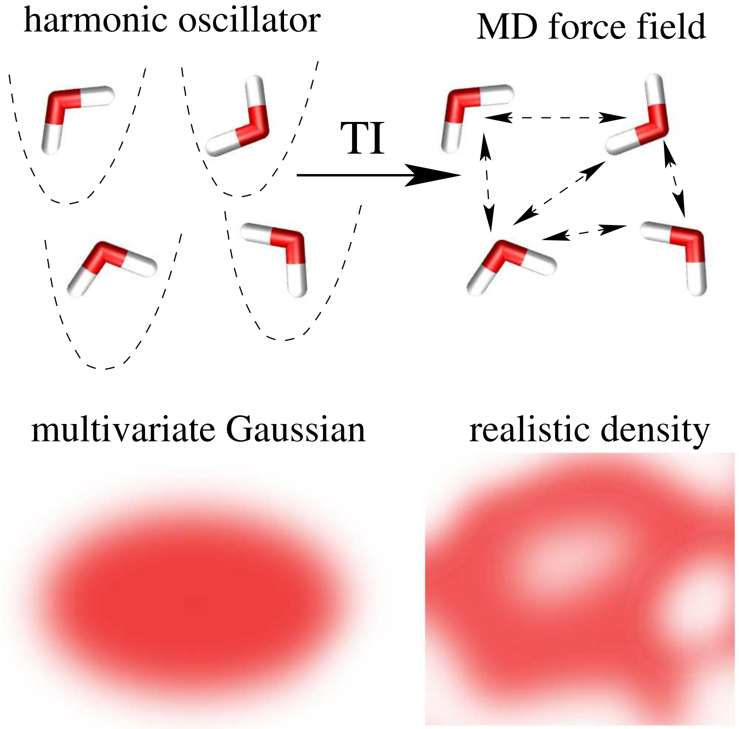

Fig. 6.2: The entropy estimation can be refined by a thermodynamic integration. The system is first simulated in a multivariate harmonic oscillator. Then the force–field parameters is gradually switched on. The entropy change during the switch is computed by TI.

one may summarize the solvent and interaction contributions to the Hamiltonian in a **potential of mean force** which depends only on the protein coordinates

$$H_{\mathrm{PMF}} = -k_B T \ln \int e^{-\beta(H_I + H_s)} d^3x_s d^3p_s \tag{6.2}$$

such that

$$\rho_p = \frac{e^{-\beta(H_p + H_{\mathrm{PMF}})}}{Z}.$$

Note that H_{PMF} is the free energy of the solvent subspace for a fixed position of the protein. By reducing the Hamiltonian to the protein subspace, the TI entropy formula (2.19) becomes applicable. Denoting the thermal average in the protein subspace by $\langle \cdot \rangle_p$ we obtain

$$\frac{dS_P}{d\lambda} = \frac{1}{k_B T^2}\left[\langle (H_S + H_{\mathrm{PMF}})(H'_S + H'_{\mathrm{PMF}}) \rangle_p - \langle H_S + H_{\mathrm{PMF}} \rangle_p \langle H'_S + H'_{\mathrm{PMF}} \rangle_p \right]_\lambda \tag{6.3}$$

H_{PMF} is difficult to access in a simulation since to evaluate it at one point $\vec{x}_p$ in the protein subspace, one has to sample over the whole solvent subspace in order to perform the integration in Eq. 6.2. The solvent subspace however is too high–dimensional to allow for such an approach.

Some appearances of H_{PMF} can be removed from Eq. 6.3 by observing that for an arbitrary quantity x

$$\begin{aligned} &\langle (H'_S + H'_{\mathrm{PMF}})x \rangle_p = \\ &= \frac{1}{Z}\int d^3x_p d^3p_p \; x \; e^{-\beta H_P} \int e^{-\beta H_{IS}} d^3x_s d^3p_s \frac{\int H'_{IS} + H'_S e^{-\beta H_{IS}} d^3x_s d^3p_s}{\int e^{-\beta H_{IS}} d^3x_s d^3p_s} x = \\ &= \langle x H' \rangle \end{aligned} \tag{6.4}$$

where the last average is taken with respect to the whole space. With this relation Eq. 6.3 becomes

$$\frac{dS_P}{d\lambda} = \frac{1}{k_B T^2}\left[(\langle H' H_S \rangle - \langle H' \rangle \langle H_S \rangle) + (\langle H' H_{\mathrm{PMF}} \rangle - \langle H' \rangle \langle H_{\mathrm{PMF}} \rangle) \right]$$

where all thermal averages are taken with respect to the whole space. Unfortunately we cannot think of a similar approach to remove the remaining appearances of H_{PMF}. So the computation of the entropy change in a subspace seems to be infeasible even though in principle it should be possible.

6.3 Computation of absolute energies in a thermodynamic integration

The most common application of TI is the computation of free energy differences. In contrast, entropy differences are calculated rarely, since these calculations do not converge as well as in the free energy case. It should also be possible to compute energy differences by TI, however we are not aware of this being widely used. Energies are more often computed directly by taking the time average over a sufficiently long trajectory. This approach, however, also suffers from sampling problems.

Consider now the case that one seeks to compute the free energy difference between an initial ($\lambda = 0$) and a final ($\lambda = 1$) state of a system by a thermodynamic integration. If one is additionally interested in the energy of the system at a fixed λ, an additional simulation at this fixed λ is necessary to obtain it
In this section we propose a way to circumvent this problem by extracting information about the energy at a fixed λ from the unphysical ensemble elements sampled by a TI that switched from $\lambda = 0$ to $\lambda = 1$.

The ensemble generated at the coupling parameter λ of a TI run approximates the density

$$\rho(\lambda) = \frac{e^{-\beta H(\lambda)}}{Z_\lambda}$$

This density can however be used to calculate ensemble averages at an arbitrary fixed λ between 0 and 1. To see this, let us without loss of generality focus on the case that one is interested in the energy of the final state, so $\lambda = 1$. Then

$$\begin{aligned} \langle\cdot\rangle_{\lambda=1} &= \int_\Gamma d\Gamma(\cdot)\frac{e^{-\beta H(1)}}{Z_1} = \frac{Z_\lambda}{Z_1}\int_\Gamma d\Gamma(\cdot)e^{\beta(H(\lambda)-H(1))}\frac{e^{-\beta H(\lambda)}}{Z_\lambda} = \\ &= \frac{Z_\lambda}{Z_1}\left\langle e^{\beta(H(\lambda)-H(1))}(\cdot)\right\rangle_\lambda . \end{aligned} \tag{6.5}$$

Note that for the special case of linear interpolation ($H(\lambda) = \lambda H$), the system sampled at the point λ in a TI is equivalent to the system at $\lambda = 1$ simulated at a temperature T/λ. In this sense thermodynamic integration is similar to simulated annealing [11].

To benchmark the proposed approach, we tested it with a simple test system. We restricted ourselves to a one–dimensional system moving along a coordinate $x \in [-10 : 10]$ in the double–well potential

$$V_A(x) = x^4 - Ax^2 + \frac{A^2}{4}$$

coupled to a heat bath with temperature T. Furthermore we assumed periodic boundary conditions. The system was simulated using Metropolis Monte–Carlo [33] for various values of the barrier height A ranging from $A = 0$ to $A = 100$. For each value we performed 20 simulations of 50000 steps each in the full Hamiltonian ($\lambda = 1$) and 20 simulations during which the Hamiltonian was linearly ramped up to simulate a TI ($H(\lambda) = \lambda H$).

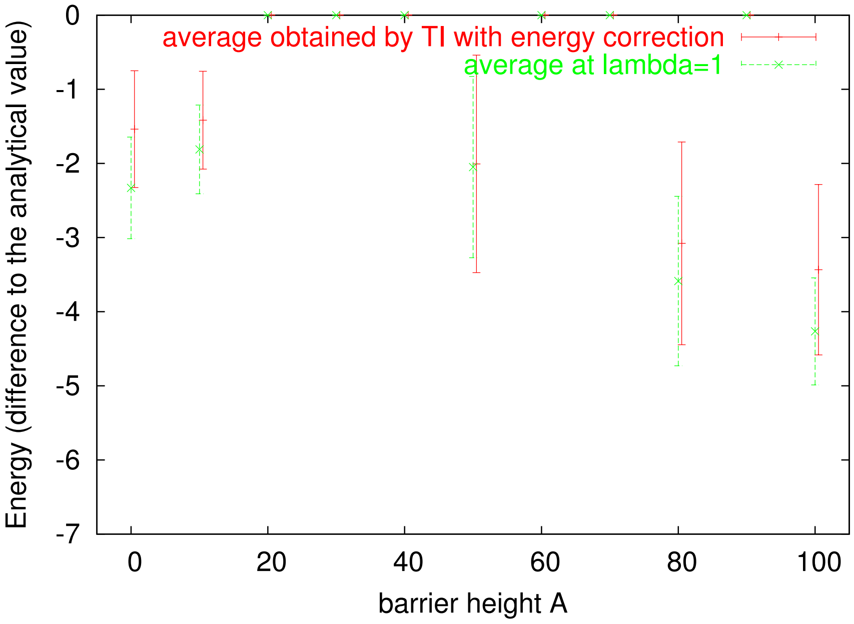

Fig. 6.3: Energy estimations for a one–dimensional double well potential. The green values was obtained by averaging over a simulation at the full Hamiltonian. The red values were obtained from a thermodynamic integration along the path $H(\lambda) = \lambda H$ by applying the proposed correction (Eq. 6.5).

The first series of simulation was used to calculate the average energy $\langle H \rangle_{\lambda=1}$ in the conventional way, i.e., by taking the time average of H in a simulation at $\lambda = 1$. The same average was computed from the TI series by correcting for the difference between the Hamiltonians $H(\lambda)$ and $H(1)$ by means of Eq. 6.5. From the resulting 20 values, the mean values and the errors of both approaches can be computed. The results are shown in Fig 6.3. Since the exact value of the mean energy can be computed in the case of the double well potential, we use this value as a reference and plot the deviation of the estimations from the analytical result.

The fact that all estimations systematically underestimate the energy is most likely due to an artefact of the Metropolis Monte–Carlo algorithm. Since the probability for the system to hop from x to $x + \Delta x$ is determined from the energy difference $H(x + \Delta x) - H(x)$, the system can tunnel through a potential barrier between $H(x)$ and $H(x + \Delta x)$. This happens at the periodic boundaries of our system, since here the periodic images of the double well potential are attached to each other and form a narrow energy peak.

The error of the proposed method is roughly as large as the error of the classical method. Thus the proposed method does not lead to an improvement as far as the accuracy is concerned. Computing the energy along with a TI is however a significant advantage from the point of view of computational efficiency, since no

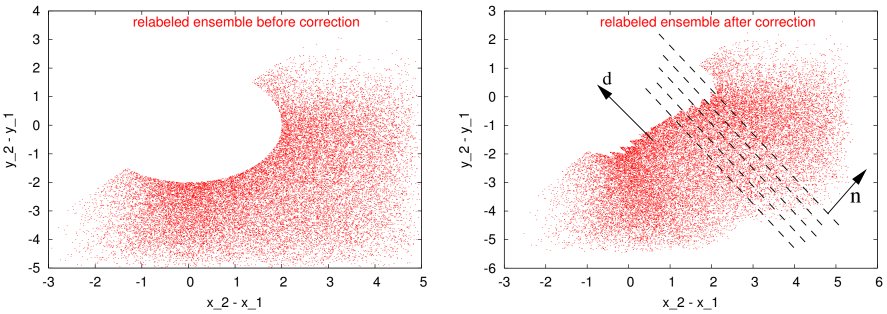

Fig. 6.4: Sketch of the slice transformation. The relabeled ensemble is projected into a relative subspace. The phase space is cut into slices which are moved such that the density becomes more similar to a Gaussian distribution.

further simulation is needed to obtain $\langle H \rangle$.

6.4 Transformations of the density

Finding a fit function which approximates the complex water density proves difficult as was shown in the previous sections. A complementary approach consists in transforming the relabeled ensemble such that it can be fit more easily. The aim of this section is to look for a transformation in configurational space $f : R^{3N} \rightarrow R^{3N}$ such that the entropy of the transformed ensemble $f(\vec{x}(t_i))$ can be estimated more accurately than in the case of the untransformed ensemble $\vec{x}(t_i)$. In fact, the introduction of polar coordinates in section 6.1.2 is a special case of this approach.

As was shown in section 6.1.2, transformations which compress or expand the density are difficult to handle, since they change the entropy. To correct the result for the entropy change induced by the transformation requires additional work which is computationally expensive.
We therefore focused on transformations which leave the entropy invariant. Such transformations can be applied arbitrary often and can thus be used to construct an iterative procedure to improve the quality of the fit. The Gaussian distribution is easy to handle numerically and seems to be applicable to a wide range cases. So we decided to choose it as fit function and look for an entropy–conserving transformation which transforms a given density into a Gaussian distribution.

6.4.1 A simple entropy–conserving transformation

As a first example, we again studied the 3x2D–example of section 4.2.4. We chose the transformation depicted in Fig 6.4. This transformation cuts the configura-

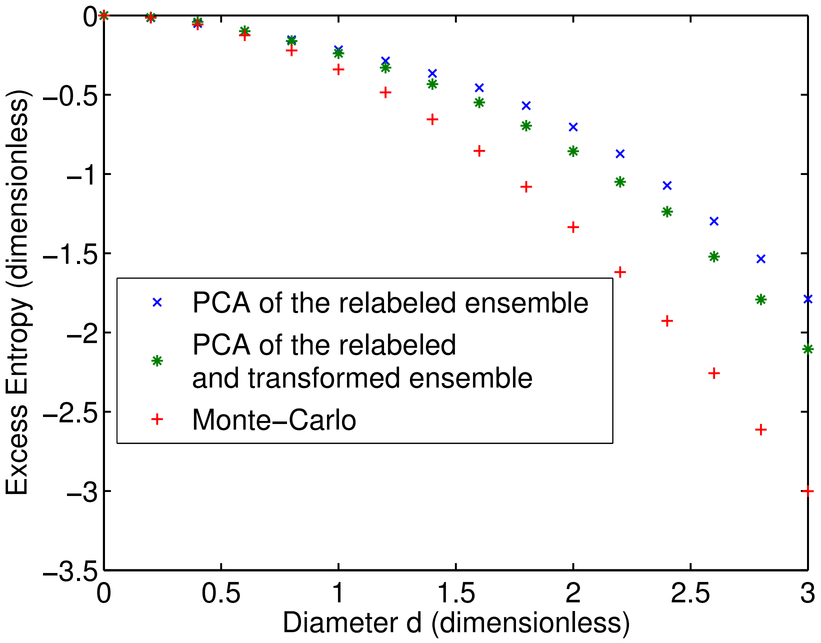

Fig. 6.5: Excess entropies in the 3x2D example. The blue and red curve are the same as in Fig. 5.4, whereas the green curve is the entropy estimation after applying the transformation of Fig 6.4 to the ensemble.

tional space into slices with some normal vector $\vec{n}$ and displaces these slices along vectors $\vec{d}$ perpendicular to $\vec{n}$. This transformation clearly conserves the entropy since according to Cavalieri's principle it does neither compress nor expand the phase space density.

Our aim is to transform the density of the relabeled ensemble into Gaussian shape by means of arbitrarily many successive transformations of this kind. We showed in section 4.2.4 that the density looks maximally non–Gaussian when it is projected into the relative subspaces. It is therefore natural to choose $\vec{n}$ such that the transformation acts in these subspaces (see Fig. 6.4 for an illustration). To transform it into a more Gaussian shape, it is clearly desirable to "close the hole" at the border. We therefore computed the mean of the ensemble elements inside each slice and then moved the slices parallel to each other such that these mean-points line up along the vector $\vec{n}$ (Fig 6.4). We performed this transformation successively in randomly picked relative subspaces. After 10 of these transformations, the entropy of the transformed density was estimated by running a PCA on the transformed ensemble and using Eq. 2.14.

The resulting entropy estimation is shown in Fig. 6.5. The approximation becomes slightly better, it remains however far from exact.

6.4.2 Continuous entropy–conserving transformations

We tried to improve this method by switching to continuous transformations. A continuous transformation can be composed of infinitesimal transformations along a vector field $\vec{v}(\vec{x})$. Again we require the transformation to conserve the entropy of the density. Trivially, this requirement is fulfilled if one imposes the even stronger condition that the density is locally conserved, i.e., that during a continuous transformation parameterized by t

$$0 = \frac{\mathrm{d}\rho}{\mathrm{dt}} = \frac{\partial \rho}{\partial \mathrm{t}} + \vec{v} \cdot \nabla \rho$$

together with the continuity equation $\frac{\partial \rho}{\partial \mathrm{t}} + \nabla \cdot (\rho \vec{v}) = 0$ this yields

$$\nabla \cdot \vec{v} = 0$$

This is the familiar condition for $\vec{v}$ to be the velocity field of an incompressible flow (i.e., an Eulerian fluid) [14]. A continuous entropy–conserving transformations can thus be constructed from every divergence–free vector field.

The problem is now to find a sequence of divergence free vector fields which generates a sequence of transformations that transform the density into Gaussian shape. Since there is no obvious choice for these vector fields, we suggest to employ a Monte–Carlo method to find an appropriate one. Such a method should generate random fields, rate them according to a criterion defining the improvement of the fit and finally apply only those vector fields, which actually lead to an improvement. We developed a method implementing this scheme, whose details are as follows:

In the first step of the algorithm one has to construct a random divergence–free vector field. To accomplish this task we used the fact that the divergence–free vector fields form a vector space: linear superpositions of divergence–free vector fields are divergence–free vector fields again. A convenient way to create random fields is thus to choose a basic set of fields and generate random divergence–free vector fields by constructing linear superpositions of these basic fields with random coefficients.

We chose our basic set to consist of vortices like sketched in Fig 6.6. The velocity field of such a vortex with center $\vec{x}_0$ and rotation axis $\vec{n}$ is given by

$$\vec{v}(\vec{x}) = f(\|\vec{x} - \vec{x}_0\|) \cdot \vec{n} \times (\vec{x} - \vec{x}_0)$$

The decay function f was chosen to be $f(r) = 1/r$, so the center $\vec{x}_0$ and the normal $\vec{n}$ uniquely define an element of the basic set. The velocity field of this basic set can be constructed at any point without dividing the space into a grid. This is an important advantage if the method is used in a high dimensional space.

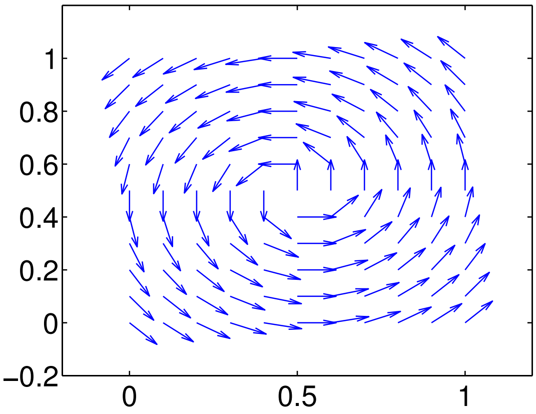

Fig. 6.6: Sketch of a vortex. We construct a divergence–free vector field by linearly superimposing several of these vortices.

In our method we construct a divergence–free vector field from this basic set in the following way: We randomly pick two elements of the ensemble under study. Then we group four vortices around them such that the velocity vectors at the points' positions point towards the center of the ensemble (Fig 6.7). By this particular choice, the two points are in average in shifted together if they are repeatedly moved along such a velocity field. In doing so we aim at condensing the trajectory in a small region of the phase space. We introduce this bias, since we know from our previous results that localized ensembles can be well fit by Gauss functions whereas the fit gets worse for inhomogeneous ensembles which are spread over a large region of the phase space.

The next step is to rate the such generated divergence–free vector field $\vec{v}$ and to have an MC algorithm decide how long it is applied to the ensemble. To rate the field we compute the entropy of the ensemble by a PCA before and after a small shift along the field. We derive the rating criterion from the fact that, according to [40] the entropy estimations based on Gaussian functions are upper bounds for the entropy of the ensemble. Since the transformation along $\vec{v}$ conserves the entropy, we can assume the fit quality to have improved if the entropy estimation yields a lower value after the transformation than before. The entropy difference ΔS of the ensemble before and after the transformation is thus a measure for the improvement of the fit.
To decide about how long the transformation is applied to the ensemble, we use the Metropolis Monte–Carlo criterion [33]. $\vec{v}$ is applied for one small time step dt if $e^{-\frac{\Delta S}{T}}$ is greater than a random number between 0 and 1. This procedure of rating, deciding and moving is repeated until the Metropolis criterion fails or a certain limiting number of steps has been carried out. Then a new divergence–free vector field is constructed. However different criteria are possible as well. For

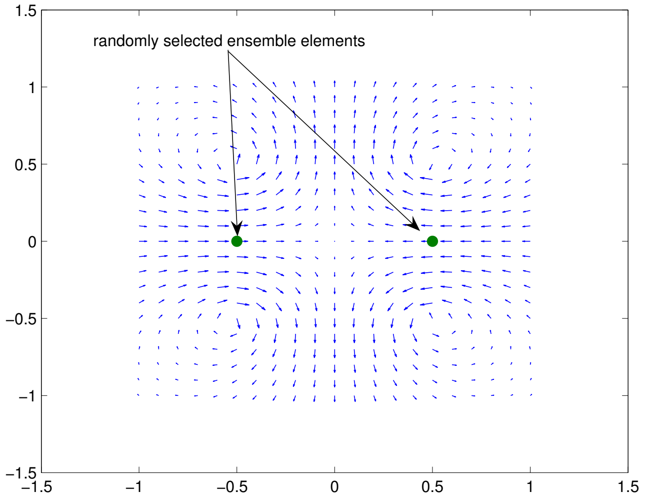

Fig. 6.7: The vector field generated by our method: We randomly pick two ensemble elements and group four vortices around them such that they are moved towards the center of the ensemble.

example, it may be useful to construct a new divergence–free vector field after every step.

To test this approach, we used a 2–dimensional test system like depicted in Fig 6.8. The ensemble was distributed in two distinct squares of different densities. We then used the proposed method to transform this ensemble into Gaussian shape. This particular example presents a worst case scenario for the method, since the ensemble is not well localized. The method has to move the squares together in an entropy–conserving way in order to render the Gaussian fit function applicable.

We used two slightly different versions of the method, one of which generates a new vector field in every step, the other of which uses one vector field as long as it improves the estimation, at most however 15 steps long. Both methods succeed in compactifying the density and improving the entropy estimation, as can be seen in Fig 6.9 and in the resulting entropy estimations which are shown in Fig. 6.10.

Both variants of the methods suffer from the same problem: The flow always contains vortices, which produce “foam” at the border of the density. This effect worsens the entropy estimation. In principle, one should expect the effect to be more pronounced when the density is not regenerated in every step, however both methods in 6.10 show the same performance.

Both approaches clearly lead to an improvement of the fit. A lot of more work

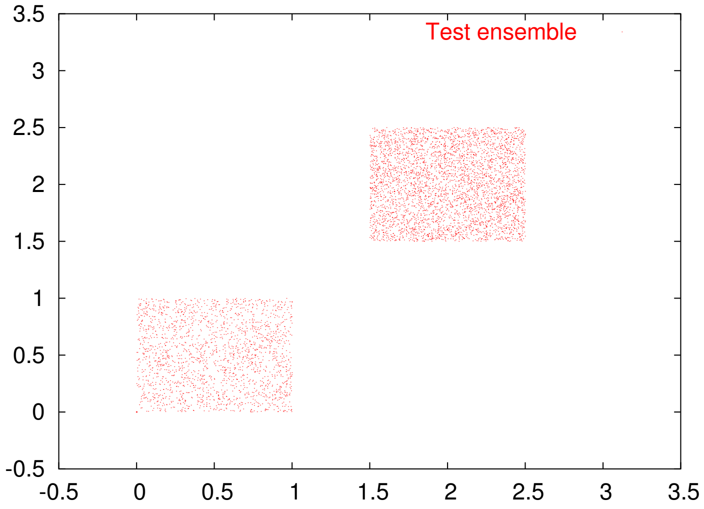

Fig. 6.8: The test case for our method: We generate a 2–dimensional ensemble of 10000 points distributed over two squares of uniform density. The lower square contains 4000 points the upper one 6000. By running the proposed method on this data we try to transform this ensemble into Gaussian shape.

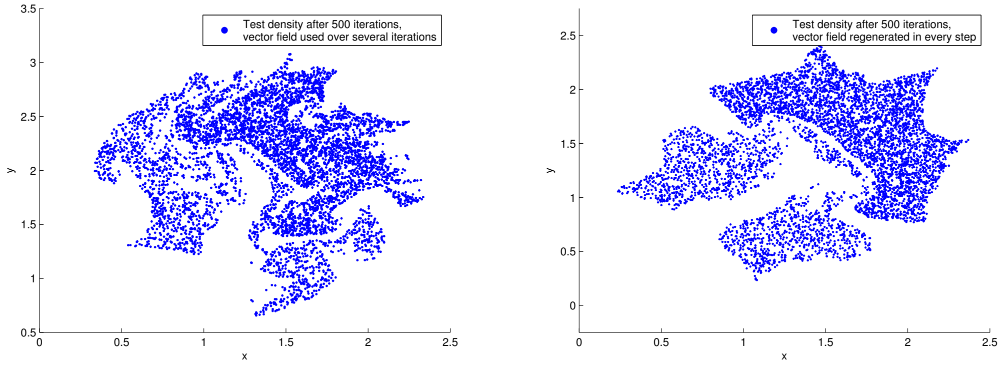

Fig. 6.9: Snapshots of the density of Fig. 6.8 after 500 iterations of the proposed method. In the right plot, the velocity field was regenerated in every step whereas in the left plot it was kept constant for several steps.

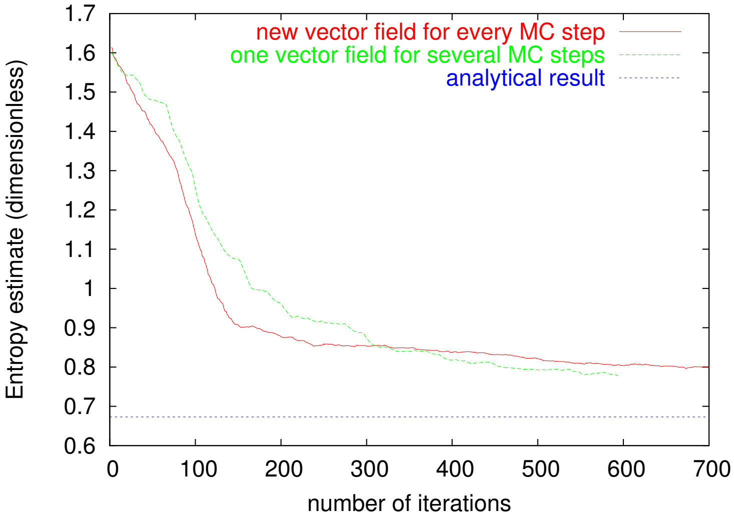

Fig. 6.10: Improvement of the entropy estimation during a run of the proposed method. The entropy estimation is enhanced, however the rate of improvement slows down for long simulation times.

however would have to be done to extend them to higher dimensions in an efficient way and to increase their overall efficiency. Also one might seek to relax the present bias and have more steps in the method done by generic methods like Monte–Carlo.

7 Summary and Conclusion

We have presented a novel method to compute solvent entropies from MD simulation trajectories. The basic idea is to “proteinize” the solvent in a trajectory by relabeling its molecules such that the resulting configurations are localized in a small region of the phase space, ideally in the vicinity of a minimum of the energy surface. Since such a localized density resembles the equally well localized density of a protein, this method renders the various entropy estimation methods developed for proteins applicable to the solvent.

We implemented this idea and applied it to several test systems in order to identify its strengths and weaknesses. The test systems included a wide range of typical cases, namely a van–der–Waals gas, a highly simplified 2–dimensional model gas and a box of water molecules.

For the van–der–Waals gas, we studied the entropy of a pure gas as well as the solvation entropy of an additional Lennard–Jones particle in the gas for varying strengths of the interparticle potential. We compared the entropy estimate obtained by an established estimation method before and after the proposed relabeling method was applied to the simulation trajectory. In both cases, the relabeling method significantly improves the entropy estimate. We also compared both of these estimates to a nearly exact reference calculation of the entropy. We find that for varying interaction strengths the entropy estimate of the relabeled trajectory reproduces all essential features of the resulting entropy change correctly. In contrast, many of them are not recognized when the entropy estimate is computed from the unrelabeled trajectory. However, both entropy estimates are found to deviate from the exact result.
We also used the van–der–Waals gas system to analyze the convergence behavior of our method. The entropy estimate of the relabeled trajectory was found to converge significantly faster than for the unrelabeled trajectory. This confirms our expectation that the relabeled trajectory samples the accessible space faster than the unrelabeled one.

By studying the 2D model gas, we could fully explain the results obtained in the study of the van–der–Waals gas. We found that the relabeling in fact simplifies the topology of the system's phase space density. This effect was identified as the reason for the improvement of the density estimation. The deviation from the exact result was attributed to the fact that the relabeled density – though its topology is simpler than for the unrelabeled case – is still too anharmonic to be perfectly approximated by a Gaussian fit function.

The study of the water system shows that our method is capable of separating the translational motion of the molecules from their rotational motion. While our method should be able to treat the entropy contribution of the first, it would have to be extended to include rotational degrees of freedom in order to handle the contribution of the latter.

In the second part of this work, we looked for ways to further improve the accuracy of the entropy estimation provided by our method. We investigated whether an improvement can be achieved by purely analytical means, such as a transformation into another coordinate system. Using the 2D model gas, we demonstrated that an improvement is possible but that this approach is limited to very small systems.

An intriguing way to further improve the accuracy of the entropy estimate is to combine the proposed technique with other entropy estimation methods. To this aim we considered several approaches, namely thermodynamic integration and entropy–conserving transformations. We find that both methods should be capable of improving the entropy estimation. Further work however would have to be done to develop a full-scale implementation of these ideas.

In all of this thesis, the relabeling procedure was applied after the MD simulation had been finished. This restriction is however by no means imperative. The solvent molecules can also be relabeled during the simulation, which opens up new possibilities. Since applying the relabeling technique prevents the solvent molecules from diffusing, the method could be used to treat different water molecules at different levels of detail. For example, the molecules nearest to the active site of an enzyme could be treated quantum–mechanically. Conversely, one might consider implementing a semi-explicit water model, which treats only the most important water molecules in an all–atom way and switches to an implicit description in other regions of the simulation box.

8 Acknowledgments

I wish to thank Dr. Helmut Grubmüller for accepting me into his research group, the challenging topic of this work, the excellent tuition and the constant support during the preparation of this thesis.

My colleagues for the great working atmosphere and the lots of fun in and outside work.

Especially Oliver Lange for providing the original source code of "g_permute" and Martin Stumpe for helpful comments on this thesis.

My Parents for their constant encouragement and their scientific as well as general advice.

I am grateful to the Freistaat Bayern for financial support by a grant according to the "Bayerische Begabtenförderungsgesetz (BayBFG)".

A Documentation of g_permute

g_permute is called from the command line. It reads a trajectory produced by an MD simulation and relabels the molecules as was described in section 3 of this thesis. Most of its input is specified by a number of command line arguments which are specified as follows:

–f filename of the trajectory to relabel. All file formats used used in Gromacs are supported.

–s filename of the topology file which was used to generate the trajectory

–o filename of the relabeled trajectory. If a file with the same name already exists, it will be overwritten by the relabeled trajetory.

–n an index file containing index groups which specify the atoms used for the calculation of the cost matrix and the relabeling (see below)

–r filename of a reference structure. This argument is optional. If it is missing, the first frame of the trajectory will be taken as reference structure.

–ms number of atoms in a solvent molecule. (3 for SPC water, 4 for TIP4P).

When run, g_permute will ask for two groups from the index file, one for the distance calculation and one for the relabeling.

The first group contains the indices of the atoms used to build the cost matrix (Eq. 3.4). Exactly one atom of each molecule must be contained in this group. The atom indices of this group must be ordered in ascending order. In case that water is used as solvent, we recommend to create a group of all water oxygen atoms and use this for the distance calculation.

The second group (called the "solvent group" in the program) contains all atoms of the solvent. In the case of SPC water, it should consist of all H and O atoms in the system being part of a water molecule. This group will be consecutively split into subgroups of –ms atoms each. These subgroups will be considered as the water molecules, so the indices of the solvent group also have to be ordered in ascending order.

References

[1] Ioan Andricioaei and Martin Karplus. On the calculation of entropy from covariance matrices of the atomic fluctuations. *J. Chem. Phys.*, 115(14):6289–6292, 2001.

[2] H. J. C. Berendsen, J. P. M. Potsma, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external heat bath. *J. Chem. Phys.*, 81:3684–3690, 1984.

[3] H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, and J. Hermans. Interaction models for water in relation to protein hydration. *Intermolecular Forces, D. Reidel Publishing Company*, pages 331–342, 1981.

[4] H.J.C. Berendsen, D van der Spoel, and R van Drunen. GROMACS: a message-passing parallel molecular dynamics implementation. *Comp. Phys. Comm*, 91:43–56, 1995.

[5] Rainer Böckmann. Molekulare Dynamik von Proteinen. *Cuvillier Verlag Göttingen*, 2002.

[6] Rainer A. Böckmann and Helmut Grubmüller. Nanoseconds molecular dynamics simulation of primary mechanical energy transfer steps in F1-ATP synthase. *Nature Struct. Biol.*, 9:198–2002, 2002.

[7] Rainer E. Burkard. Selected topics on assignment problems. *Discrete Applied Mathematics*, 123:257–302, 2002.

[8] Rainer E. Burkard and Eranda Cela. Linear Assignment Problems and Extensions. *Bericht Nr. 127, SFB Optimierung und Kontrolle*, 1998.

[9] William J. Cook. Combinatorial optimization. *Wiley-Interscience series in discrete mathematics and optimization*, 1998.

[10] Bert L. de Groot and Helmut Grubmüller. Water Permeation Across Biological Membranes: Mechanism and Dynamics of Aquaporin-1 and GlpF. *Science*, 294:2353–2357, 2001.

[11] Maurice de Koning, A. Antonelli, and Sidney Yip. Optimized Free–Energy Evaluation Using a Single Reversible–Scaling Simulation. *Phys. Rev. Lett.*, 83(20):3973–3977, 1999.

[12] Warren L. DeLano. The PyMOL Molecular Graphics System. *http://www.pymol.org.*

[13] D. Eisenberg and A. D. McLachlan. Solvation energy in protein folding and binding. *Nature*, 319:199–203, 1986.

[14] Torsten Fliessbach. Lehrbuch zur theoretischen Physik I – Mechanik. *Spektrum Akademischer Verlag GmbH, Heidelberg, Berlin, Oxford*, 1996.

[15] Helmut Grubmüller. Predicting slow structural transitions in macromolecular systems: conformational flooding. *Phys. Rev. E*, 52:2893, 1995.

[16] Jean Pierre Hansen and Ian R. McDonald. Theory of simple liquids, second edition. *Academic Press, London*, 1986.

[17] Berk Hess. Stochastic Concepts in Molecular Simulation. *Universal Press, Veenendaal.*

[18] Berk Hess. Similarities between principal components of protein dynamics and random diffusion. *Phys. Rev. E*, 62:8438–8448, 2000.

[19] Berk Hess. Convergence of sampling in protein simulations. *Phys. Rev. E*, 65, 2002.

[20] W. Humphrey, A. Dalke, and K. Schulten. VMD - Visual Molecular Dynamics. *J. Molec. Graphics*, 14:33–38, 1996.

[21] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[22] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives. Development and Testing of the OPLS All-Atom Force Field on ConformationalEnergetics and Properties of Organic Liquids. *J. Am. Chem. Soc.*, 118:11225–11236, 1996.

[23] L. A. Kaloujnine and V. I. Suscanskij. Transformationen und Permutationen: eine Einführung in die Gruppentheorie. *Deutsch Verlag*, 1986.

[24] Martin Karplus and Joseph N. Kushick. Method for Estimating the Configurational Entropy of Macromolecules. *Macromolecules*, 14:325–332, 1981.

[25] J.G. Kirkwood. Statistical mechanics of fluid mixtures. *J. Chem. Phys.*, 3:300–313, 1935.

[26] Harold W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[27] L. D. Landau and E. M. Lifschitz. Lehrbuch der theoretischen Physik – statistische Physik, 7. Auflage. *Akademie–Verlag Berlin*, V, 1987.

[28] Themis Lazaridis and Martin Karplus. Orientational correlations and entropy in liquid water. *J. Chem. Phys.*, 105(10):4294–4316, 1996.

[29] C. Levinthal. Are there pathways for protein folding ? *J. Chim. Phys.*, 65:44–45, 1968.

[30] Erik Lindahl, Berk Hess, and David van der Spoel. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Journal of Molecular Modeling*, 7(8):306–317, 2001.

[31] MagicLogic. Library to solve the linear assignment problem. *http://www.magiclogic.com/assignment.html.*

[32] Gerald Mathias and Paul Tavan. Angular resolution and range of dipole–dipole correlations in water. *J. Chem. Phys.*, 120(9):4393–4403, 2004.

[33] N. Metropolis, A. Rosembluth, M. Rosembluth, and A. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[34] Shuichi Miyamoto and Peter A. Kollman. SETTLE: an analytical version of the SHAKE and RATTLE algorithm for rigid water models. *Journal of Computational Chemistry*, 13(8):952 – 962, 1992.

[35] Gerd Ulrich Nienhaus. Physik der Proteine. *Physik Journal*, 4:37, 2004.

[36] S. Nose. A molecular dynamics method for simulations in the canonical ensemble. *Molecular Physics*, 100(1):191–198, 1984.

[37] Christine Peter, Chris Oostenbrink, and Wilfried F. van Gunsteren Arthur van Dorp. Estimating entropies from molecular dynamics simulations. *J. Chem. Phys.*, 120(6):2652–2661, 2004.

[38] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical Recipes in C, Second Edition. *Cambridge University Press*, 1988.

[39] Heiko Schaefer and Wilfried F. van Gunsteren Alan E. Mark. Absolute entropies from molecular dynamics simulation trajectories. *J. Chem. Phys.*, 113(18), 2000.

[40] Jürgen Schlitter. Estimation of absolute and relative entropies of macromolecules using the covariance matrix. *Chem. Phys. Lett.*, 215(6):617–621, 1993.

[41] James P. Stoessel and Martin Nowak. Absolute Free Energies in Biomolecular Systems. *Macromolecules*, 23:1961–1965.

[42] Hansson T, Oostenbrink C, and van Gunsteren W. Molecular dynamics simulations. *Curr Opin Struct Biol.*, 12(2):190–6, 2002 Apr.

[43] D van der Spoel, AR van Buuren, and E Apol et. al. Gromacs User Manual.

[44] Wilfried F. van Gunsteren, Paul K. Weiner, and Anthony J. Wilkinson. Computer simulations of biomolecular systems. *ESCOM Science Publishers, Leiden*, 2, 1993.

[45] W. F. van Gunsteren et. al. The GROMOS96 manual and user guide. *Hochschulverlag AG an der ETH Zürich, Zürich*, 1996.

[46] R.W. Zwanzig. High-temperature equation of state by a perturbation method. I. Nonpolar gases. *J. Chem. Phys.*, 22:1420, 1954.

Erklärung zur Diplomarbeit

Hiermit versichere ich, die vorliegende Diplomarbeit selbständig angefertigt zu haben und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

Göttingen, 22. März 2005

(Friedemann Reinhard)