Tip4p Water Model in the Ice I_h Configuration

Master Thesis

submitted by

Johannes Zierenberg

December 2010

Supervisor:
Prof. Dr. Wolfhard Janke

Referees:
Prof. Dr. Wolfhard Janke
Prof. Dr. Bernd A. Berg
(Florida State University)

Institut für Theoretische Physik
Fakultät für Physik und Geowissenschaften
Universität Leipzig

Abstract: The purpose of this study was to investigate different 4-point water models (Tip4p) inside the ordinary, hexagonal ice phase (ice I_h), with respect to the structure of their degenerate local energy minima. Another aim was to find out how the underlying structure influences the results of Monte Carlo (MC) simulations in and near the ice phase. The first aim was achieved by minimization of random water clusters in the experimentally known ice I_h configuration for different parameterizations, and characterization of the unit cells. In order to obtain representative system sizes, the long-range interactions were cutoff at 25Å. It was found that the Tip4p models have stable hexagonal minimum configurations with specific lattice constants. The results indicated that the Tip4p model already lifts the energy degeneration slightly. MC simulations of 128 molecules with fixed oxygen positions revealed a qualitative dependence of the MC results on the lattice constant and indicated that energy cutoffs above 9Å provide very similar results. Inclusion of the translational degrees of freedom was shown to be a promising approach in the search for a phase transition between the liquid and the solid phase of the Tip4p water model.

Contents

List of Figures

List of Tables

1 Introduction

Water is, without a doubt, an essential part of our world, playing an important role in many biological and chemical processes.

In addition its physical properties are quite unusual. It presents a number of "anomalous" properties [1, 2, 3, 4]. For one, water at room temperature is in its liquid phase, while molecules with similar mass remain in the gaseous phase. Moreover, liquid water is observed to be denser than ice, resulting in the fact that ice floats on water. The maximal density is achieved at a temperature of $277K$.The solid structure of water depends largely on the temperature and pressure and may take various forms. This work will focus on the ordinary ice phase (ice I_h), presenting a hexagonal structure.

The extraordinary behavior of water gives rise to the question of the underlying causes and following consequences. Therefore, a molecular understanding of the properties and structure of water is of considerable interest. For this purpose scientists developed several models throughout the last century. As a result, the mystical anomalies obtained plausible explanations. For example, the liquid phase at room temperature can be explained by the strength of the hydrogen bond between molecules, which will be discussed in detail later.

The structure and property of water has been investigated with much effort in the past. Especially the ice I_h phase is now fairly well understood (for a detailed discussion see [1]). In 1921 W. H. Bragg proposed the hexagonal arrangement of oxygen atoms [5]. Until now, the oxygen structure has been investigated extensively by X-ray, electron and neutron diffraction [6, 7]. In principle, they confirmed the tetrahedral arrangement of neighboring oxygen atoms, forming a hexagonal structure with only small deviations from the ideal arrangement. As mentioned above, several more ice phases exist, many of which are also discussed in [1].

Due to the importance of water in many biological and chemical processes, it is very important to have appropriate water models. Today, many simulations are performed in order to investigate processes on a biological or chemical scale, often involving water molecules. The search for water models has been ongoing for more than 70 years and has been summarized up to 2002 by Guillot [8]. The first realistic model for water was proposed in 1933 by Bernal and Fowler [9]. They suggested a simple electrostatic point model with its center of negative charges shifted away from the oxygen, along the dipole axis, towards the positive hydrogen charges. Barker and Watts [10] as well as Rahman and Stillinger [11] performed the first computer simulations of liquid water towards the end of the 1960s. A

number of models were developed in the subsequent decades, including a simple point charge model (SPC) [12] with almost tetrahedral geometry; the so-called Tip3p and Tip4p models proposed by Jorgensen et al. [13]; as well as the Tip5p model [14]. Usually, the parameters of the models are obtained by reproduction of thermodynamic properties in simulations. For this reason different models or parameterizations are suitable for different applications or phases. Recently it was shown that Tip4p models predict the equilibrium pressure reasonably well and are stable in the ice phase throughout the simulation [15].

This work compares different parameterizations of the Tip4p model, which is supposed to be already quite general, in the ordinary ice phase with respect to the structure of the unit cell and its stability in more detail. Furthermore it is investigated whether the ground states of the continuous models support the energy degeneration, which has been proposed [16] and experimentally verified [17]. Afterwards the effects of the obtained results on Monte Carlo results are investigated.

The thesis starts with the description of the model and the ice I_h structure known from experiment. It continues with a brief overview of the implementation of the model, followed by an explanation of the numerical methods used. Afterwards a short chapter about possible optimizations is presented, including the simulation on graphics cards. In the end the numerical results are presented and discussed.

2 Theory

2.1 Water model

When it comes to investigating the behavior of water on a computer, the molecules need to be approximated by a model. In general there are two requirements on the model. On the one side it needs to be a generic model, reproducing a large set of properties. On the other side it needs to be simple enough, in order to be computable at reasonable costs.

There exist several simple water models, for example spherical approximations, dipole models and models on atomic scale. They all try to reproduce certain macroscopic properties of water. However, none are able to reproduce all macroscopic properties in both the pure liquid phase and in solution [3], or as a matter of fact, in the solid phase. One therefore needs to choose a model, appropriate for the task at hand. The models on atomic scale differ by the number of points of interaction and their parameterization. Maybe the most intuitive model is the 3-point model, in which the oxygen as well as both hydrogen atoms are considered as point particles and their charges are localized on the corresponding positions [13]. "Higher" point models usually split and shift the charges a little bit.

In common atomic water models, two interactions are considered between water molecules, while the intramolecular interactions are neglected as rigid, uncharged molecules are assumed. For point charges the electrostatic interaction is determined by the *Coulomb potential*, which is proportional to the charges and the inverse distance. Furthermore the *Lennard-Jones potential* describes the interaction between two uncharged molecules or atoms resulting in their characteristic behavior. Despite the experimentally [18] supported presumption that the Lennard-Jones interaction is rather non-spherical, it is often simply considered as a spherically symmetric interaction between neutral molecules, with the oxygen atoms being the source of interaction. In this case the 12-6 Lennard-Jones potential is considered, including a repulsive term proportional to r^{-12} and an attractive term proportional to r^{-6}. The potential energy between two molecules a and b is thus given by the sum of the interactions. As the molecules are approximated as rigid bodies, where the internal angles and distances are fixed, the interactions are only considered between points of different molecules and not within a single molecule. In the following q_i is the charge of the interaction point i, r_{ij} is the distance between charge i and j and k_C is the Coulomb constant. Furthermore r_{OO}

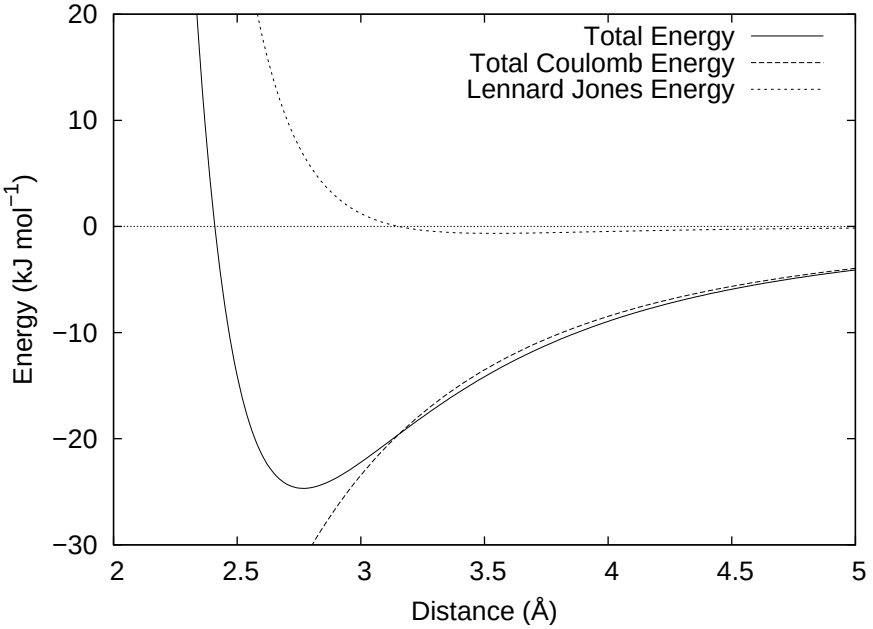


Figure 2.1: Interaction energies between two water molecules versus the distance between the oxygen atoms. The molecules are oriented in a "hydrogen-bond" configuration (see Sec. 6.1.1).

is the distance between the oxygen atoms and *tipA* as well as *tipC* are constants of the Lennard-Jones interaction. The interaction energy between two molecules is thus given by

$$E^{ab} = k_C \sum_{i}^{\text{in a}} \sum_{j}^{\text{in b}} \left(\frac{q_i q_j}{r_{ij}}\right) + \frac{tipA}{r_{OO}^{12}} - \frac{tipC}{r_{OO}^{6}}. \tag{2.1}$$

The total energy between two molecules may have a minimum, depending on their spatial orientations. Fig. 2.1 shows an example for the Lennard-Jones and Coulomb interaction terms and the resulting total energy. It is important to notice that for long ranges the Lennard-Jones potential decreases with r^{-6}, while the Coulomb potential decreases only with r^{-1}. This means that at large distances the energy contribution of the Coulomb interaction is dominating.

2.1.1 Tip4p water model

One of the simple water models is the *Tip4p* water model. Tip4p is the abbreviation for the *4-point-transferable-intermolecular-potential*, referring to a simple rigid water model with four interaction points contributing to the potential energy. It is based upon the early proposal of Bernal and Fowler [9]. As for the

Parameter	Tip3p[13]	Tip4p[13]	Tip4p/2005[19]	Tip4p/Ice[20]
Distance H-O (Å)	0.9572	0.9572	0.9572	0.9572
Distance O-charge(Å)	0.0	0.15	0.1546	0.1546
Angle (deg)	104.52	104.52	104.52	104.52
Charge H	0.417	0.52	0.5564	0.5564
$TipA$ ($\text{Å}^{12}\text{kcal}/\text{mol}$)	582×10^3	600×10^3	731.6×10^3	858.2×10^3
$TipC$ ($\text{Å}^{6}\text{kcal}/\text{mol}$)	595	610	736.2	850.8
k_C ($\text{Å kcal}/\text{mol}$)	332.064	332.064	332.064	332.064

Table 2.1: Some parameterizations of the Tip4p water model.

real molecule, the model consist of two hydrogen and one oxygen atom. The hydrogen atoms bond with the oxygen atom by "sharing" their electrons, resulting in a slightly positive charge of the hydrogen and a slightly negative charge of the oxygen. These bonds are formed at a rigid angle, resulting in a dipole.

In the Tip4p model the charge from the oxygen is shifted along the molecule's dipole axis, causing the model to have four points of interactions (see Fig. 2.2). The specific distances depend on the parameterization of the model.

Consequently the Tip4p model does not uniquely define a specific molecule, but rather a class of four-point water models with different parameters. Tab. 2.1 shows the parameterizations considered in this work. The historical convention $^{\text{cal}}/_{\text{mol}}$ can be converted into SI units by 1J = 4.183cal and the Avogadro constant.

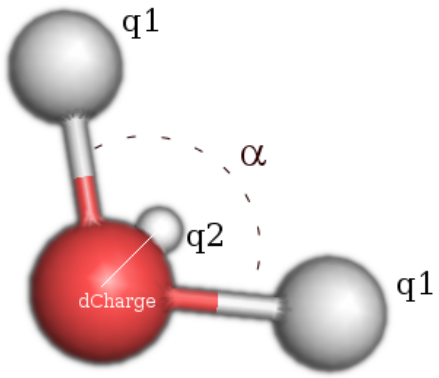


Figure 2.2: General Tip4p water model

2.1.2 Forces in the Tip4p model

In the Tip4p model the water molecule is considered to be a rigid body with discrete mass points. This means that in the center of mass frame of a single molecule no intermolecular forces are acting, such that the positions and angles are conserved. The external forces acting on the atoms in this center of mass frame cause a torque responsible for a rotation. Furthermore external forces will act on the center of mass resulting in a translation. Thus the motion of the rigid body can be divided in the translation of the center of mass and the rotation of the body. The forces are described in more detail in the following subsections.

Center of mass force

For a discrete rigid body with finitely many mass points, the force on the center of mass is identical to the force on the center of mass of a system of free mass points (see [21], p. 159). Considering a single water molecule as a system of N free particles, it is possible to separate the forces acting on each atom into *external forces* due to neighboring molecules, and *internal forces* due to internal interactions.

The equation of motion of any particle from this system contains the sum over the internal forces and the external force

$$m_i \ddot{\vec{r}}_i = \sum_{j \neq i} \vec{F}_{ji}^{(int)} + \vec{F}_i^{(ext)}, \tag{2.2}$$

where $\vec{F}_i^{(ext)}$ is the total external force acting on particle i. Furthermore, the center of mass can be considered as a mass point of mass $M = \sum_{i=1}^{N} m_i$ at position $r_{cm} = \frac{\sum_{i=1}^{N} m_i \vec{r}_i}{M}$. The corresponding equation of motion takes the form

$$M\ddot{\vec{r}}_{cm} = \sum_{i=1}^{N} m_i \ddot{\vec{r}}_i = \sum_{i=1}^{N} \sum_{j \neq i} \vec{F}_{ji}^{(int)} + \sum_{i=1}^{N} \vec{F}_i^{(ext)}.$$

The first sum will vanish due to Newtons law of actio-reactio, such that the force on the center of mass is given as the sum of all external forces on the particles of the system considered

$$M\ddot{\vec{r}}_{cm} = \sum_{i=1}^{N} \vec{F}_i^{(ext)} = \vec{F}_{cm}. \tag{2.3}$$

This means that the force on the center of mass of a water molecule is obtained by the direct sum of the external forces acting on each of the four interaction points, regardless of their mass.

Torque

The torque $\vec{T}$ causes a rotation of a mass point about an axis. It is defined as

$$\vec{T} = \vec{r} \times \vec{F}, \tag{2.4}$$

where $\vec{r}$ is the distance between the point of rotation and the mass point and $\vec{F}$ is the force acting on that point. In the case of the rigid body, the total torque about its center of mass is of interest. Considering the rigid body as a system of free particles, the torque of a single particle i is given by

$$\vec{T_i} = \vec{r_i} \times \vec{F_i} = \vec{r_i} \times \left(\sum_{j \neq i} \vec{F}_{ij}^{(int)} + \vec{F}_i^{(ext)} \right), \tag{2.5}$$

where $\vec{r_i}$ is the position vector in relation to the center of mass. The internal forces are acting along a direct path between two interacting points such that

$$\vec{T_i} = \sum_{j \neq i} \left(F_{ij}^{(int)} \frac{\vec{r_i} \times (\vec{r_i} - \vec{r_j})}{r_{ij}} \right) + \vec{r_i} \times \vec{F}_i^{(ext)}. \tag{2.6}$$

Summing up all individual contributions, the term involving the internal forces vanishes due to Newtons third law. The total torque of the rigid body is thus given by the sum of the individual torques caused by external forces

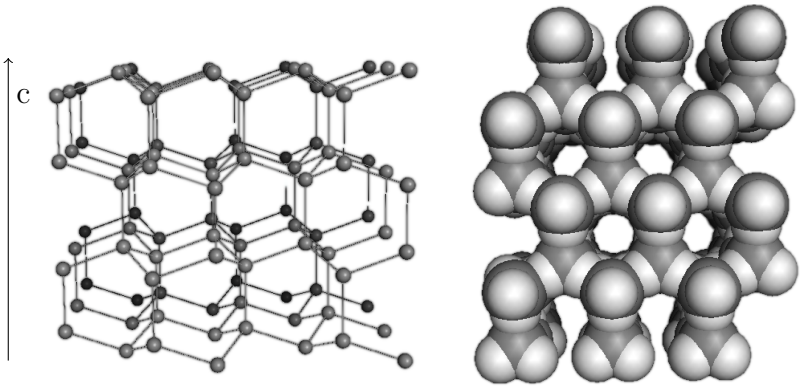
$$\vec{T} = \sum_i \vec{T_i} = \sum_i \vec{r_i} \times \vec{F}_i^{(ext)}. \tag{2.7}$$

2.2 Ice I_h phase

Compared to liquid water, ice is very well understood. Throughout the last century the structure of ice was experimentally analyzed using X-Ray, electron and neutron diffraction. This gave insight into the crystal structure, the symmetries and the acting forces. The ordinary ice known from day-to-day life is in the so called *ice* I_h phase. Furthermore there exist a number of ice phases for lower temperatures or higher pressures, which will not be considered in this work.

2.2.1 Structure

In the ice I_h phase the oxygen atoms have a crystalline structure, while the arrangement of the hydrogen atoms and thus the orientation of the molecules can be considered glass-like [9]. Every oxygen atom is at the center of a tetrahedron, which is formed by the four neighboring oxygen atoms, each a lattice constant a away [1]. Each water molecule bonds to its four nearest neighbors by *hydrogen*



(a) The ice I_h phase structure represented by the oxygen backbone.

(b) View of Ice1 structure along the c-axis. Hard sphere approximation of atoms shows the vacant shafts along the c-axis.

Figure 2.3: Structure of water in the ice I_h phase

bonds, directing its hydrogen atoms towards the neighboring molecules *lone-pairs* of electrons as well as directing each of its lone-pairs towards a hydrogen atom of a neighboring molecule. This results in a hexagonal lattice (see Fig. 2.3(a)) in which intermolecular cohesion is large [1].

In Fig. 2.3(a) a particular direction is specified, often referred to as the *c-axis*, perpendicular to which the layers are oriented. These layers consist of hexagonal rings and repeat their configuration every second layer. There are in principle two forms of hexagonal rings, one inside a layer, where opposite sites have opposed displacements from the hexagonal plane (sometimes called "chair" configuration) and one between two layers, where opposite sites have the same displacement from the hexagonal plane (referred to as "boat" configuration) [22]. Mapping the chair configuration to a plane, each angle is 120° and all sides have the same length.

When looking along and perpendicular to the c-axis *vacant shafts* are visible. The resulting perforated structure is an important characteristic and responsible for the fact that ice floats on liquid water [1]. This is shown in Fig. 2.3(b), by representing the atoms with their spacial occupation in order to illustrate the perforated structure. A schematic representation is given in Fig. 2.4

In general a crystal structure is made up of small repeating units, reproducing any other part of the point lattice by translations. They are called *unit cells* and can be composed of a single point or a set of points. The unit cells fill the whole lattice space and are symmetric under translation by the lattice vectors (for further information see [23]). For the ice I_h crystal the unit cell consists of four molecules. One possible unit cell is shown in Fig. 2.5.

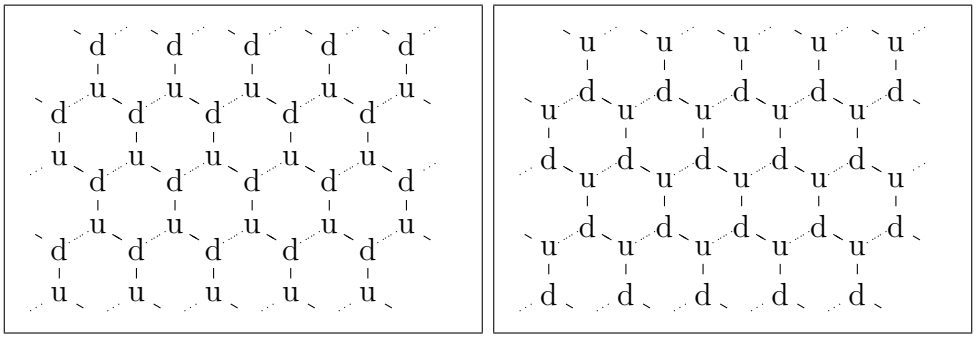


Figure 2.4: Projection of the alternating layers into the x-y plane. The up(u) sites are above and the down(d) sites are below the plane.

In solid state physics, crystal structures are characterized by their symmetries. It is commonly accepted that the ice I_h crystal has the symmetry $P6_3/mmc$ [1]. This is the name of a non-cubic space group in the international nomenclature. Converted to Schoenflies nomenclature this is read D_{6h}. This means that the unit cell is symmetric under a rotation about the c-axis by $2\pi/6$ [23]. Furthermore, there exist three perpendicular axes, about which a rotation by $2\pi/2$ is a symmetry operation, as well as a mirror plane perpendicular to the first rotation axis. Due to the regular displacement from the hexagonal plane, the rotational symmetry about the c-axis only holds for rotations of $2\pi/3$. It is important to notice that, depending on the choice of the unit cell, the symmetry can be obvious or hidden. It is most obvious choosing a full hexagonal ring over three layers.

The basis vectors of the lattice, by which the unit cell can be periodically trans-

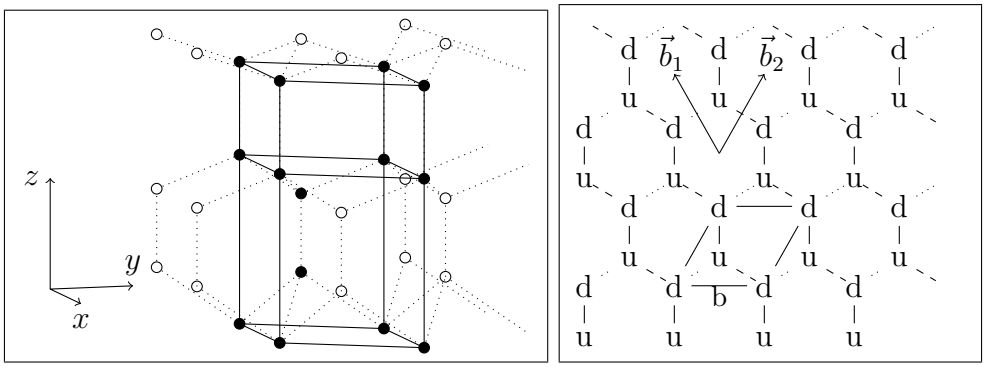


Figure 2.5: Possible unit cell of water in the ice I_h phase, 3D (black sites are inside unit cell) and 2D (with basis vectors)

lated, are given by

$$\begin{aligned}\vec{b}_1 &= \frac{1}{2}b\hat{x} - \frac{1}{2}\sqrt{3}b\hat{y} \\ \vec{b}_2 &= \frac{1}{2}b\hat{x} + \frac{1}{2}\sqrt{3}b\hat{y} \\ \vec{b}_3 &= c\hat{z},\end{aligned} \tag{2.8}$$

where $\hat{x}, \hat{y}, \hat{z}$ are the unit vectors in Cartesian coordinates (see Fig. 2.5). In the literature b is often called a, which is not done in this context since a is defined to be the distance between nearest neighboring oxygen atoms, the lattice constant.

There have been some uncertainties with respect to the unit cell dimension, as emphasized by Lonsdale in 1958, because the measured ratio of the unit cell side length c/b (see (2.8)) seems to be smaller than the value of a structure build from perfect tetrahedra [24]. Lonsdale proposed either a shorter hydrogen bond along the c-axis, or a difference in the O-O-O angles from the perfect tetrahedral angle 109.47°, or a combination of both, supported by following studies [7]. Measurements using synchrotron radiation verified this deformation [25] and identified it with the O-O-O angles being wider when two oxygens are in the hexagonal plane. The lattice constants were identical within the limit of the error.

The placement of the hydrogen atoms in the ice I_h phase is even more complicated than in the case of the oxygen atoms. Because of the similarity of several physical properties of ice to those of water vapor, Bernal and Fowler [9], as well as Pauling [16], reasoned that the H_2O molecules have to be intact. Furthermore the following *ice rules* have to be fulfilled:

Ice rule 1: Between two adjacent oxygen atoms lies exactly one hydrogen atom, closer to one of the oxygen atoms. This bond is then called *hydrogen bond*.

Ice rule 2: To each oxygen atom belong exactly two hydrogen atoms, which are closest to it.

These ice rules are the basis of argumentation for a residual entropy by Pauling, considered in the following subsection. With the possibility of neutron diffraction, Peterson and Levy essentially confirmed this for low temperatures [6].

The water molecule structure is not very different from isolated molecules. Eisenberg and Kauzman state that the distance between the oxygen and the hydrogen atom is about 1.01Å[1], which deviates from the value obtained for the Tip4p model. Furthermore they state that the H-O-H angle is probably not much greater than the angle of isolated molecules.

2.2.2 Residual entropy

The hydrogen arrangement in the ice I_h phase is not unique, considering the ice rules. Pauling argued in 1935 that all arrangements which satisfy the ice rules, assuming intact molecules, have the same probability to occur and are degenerate. This implies that a crystal satisfying these three conditions is not completely ordered at zero temperature, thus having a *residual entropy*, a non-vanishing positive entropy term S_0 at zero temperature. The residual entropy is given in terms of the Boltzmann factor k and the number of configuration for N molecules W_N by

$$S_0 = k \ln(W_N) > 0. \tag{2.9}$$

Pauling calculated the residual entropy by estimating the number of configurations in two different ways [16]. There are six ways a water molecule can orient itself in a tetrahedral structure satisfying ice rule 2. If furthermore all molecules are given one of those orientations at random, each bond has two adjacent molecules with the probability of 1/2 each that one of their hydrogen atoms lies along that bond. Thus there is a 1/4 chance to satisfy ice rule 1, having exactly one hydrogen atom between all two oxygen atoms. This results in a total number of configurations of

$$W_1^{(Pauling)} = (6/4)^N = (3/2)^N. \tag{2.10}$$

The same result is obtained by the following equivalent argument. Ignoring ice rule 2, simply demanding one hydrogen atom between two bonding oxygen atoms there are 2^{2N} configurations ($2N$ is the number of bonds), each hydrogen atom having the choice of being closer to one of the two oxygen atoms. Considering any oxygen atom, there are hence $2^4 = 16$ possible arrangements of the hydrogen atoms, ten of which are ruled out by ice rule 1, because they result in $(H_4O)^{++}$, $(H_3O)^+$, $(OH)^-$ or O^{--}. Thus for each oxygen atom only $6/16 = 3/8$ configurations are allowed resulting in a total number of configurations of

$$W_2^{(Pauling)} = 2^{2N}(3/8)^N = (3/2)^N. \tag{2.11}$$

With the use of the definition of the residual entropy this result leads to

$$S^{(Pauling)} = kN \ln(3/2) = R \ln(3/2), \tag{2.12}$$

where the molar gas constant is given as $R = 8.314472(15)^{\mathrm{J}}/_{\mathrm{mol\ K}}$ [26] . This results in a residual entropy of

$$S^{(Pauling)} = 3.3712...^{\mathrm{J}}/_{\mathrm{mol\ K}} = 0.80574...^{\mathrm{cal}}/_{\mathrm{deg\ mol.}}{}^{1} \tag{2.13}$$

[1]The conversion to historical units requires K → deg and 4.184J = 1cal.

Refined estimations [27] of the residual entropy by means of multicanonical simulations give

$$S^{(MUCA)} = (0.81550 \pm 0.00021)^{\mathrm{cal}}/_{\mathrm{deg\ mol}}, \tag{2.14}$$

in good agreement with the results of the series expansion method by Nagle [28]. With the use of calorimetry Giauque and Stout [17] obtained an experimental value for the residual entropy in good agreement with Pauling and the theoretic estimation.

$$S^{(exp)} = (0.82 \pm 0.05)^{\mathrm{cal}}/_{\mathrm{deg\ mol}} \tag{2.15}$$

This shows that each possible ice state has almost the same probability.

Neutron diffraction experiments done by Peterson and Levy confirmed this disordered structure for -50 and -150°C, analyzing D_2O ice [6]. This degeneration of ground states is essentially responsible for the previously mentioned glass-like behavior of the water. It may be argued that, due to the different possible arrangements of the dipoles, there exist ground states that are more stable than others [29]. For low temperatures, the ordered form of ice I_h (ice XI) was experimentally found to exist for KOH-doped ice [30], even though the energy differences to the local minima of ice I_h are very small [31].

3 Implementation of the Model

3.1 Construction of the periodic ice structure

In Sec. 2.2 the structure of ice I_h was discussed. Reproducing this structure, it is important to keep in mind the hexagonal layers perpendicular to the c-axis (see Fig. 2.4) and the tetrahedral arrangement. These are two criteria that help to determine the Cartesian dimensions in terms of a given lattice constant a, the distance between two bonding oxygen atoms.

Projecting the layer to a plane results in a hexagon with side lengths of a_{xy} and inner angles of 120° each. This is schematically shown in Fig. 3.1. Furthermore there are three variables introduced: dx, dy and dz, the displacement from the hexagonal plane indicated by up(u) and down(d). With basic geometry it is possible to express the variables in terms of a_{xy}:

$$\begin{aligned} dx &= a_{xy}\cos(30°) = \frac{\sqrt{3}}{2}a_{xy} \\ dy &= a_{xy}\sin(30°) = \frac{1}{2}a_{xy} \\ (2dz)^2 &= a^2 - a_{xy}^2 \end{aligned} \tag{3.1}$$

The last equation presents a_{xy} as a function of a and dz. A unique solution of a_{xy} and dz in terms of a is not possible without two linearly independent equations. As mentioned above, a second criterion is available, namely the fact that the oxygen atoms are arranged tetrahedral, such that the angles are 109.47°. For the determination of the variables any tetrahedral angle in the system can be considered as reference. In the following argument, the angle between a given molecules neighbor along the y-axis and the neighbor counterclockwise in the layer plane was chosen (see Fig. 3.1). Knowing the length of each vector is a, the following condition is obtained

$$\begin{aligned} \vec{v}_1 \cdot \vec{v}_2 = a^2\cos(109.47°) &= (0, -a_{xy}, 2dz)\,(dx, dy, 2dz) \\ &= -\frac{1}{2}a_{xy}^2 + 4dz^2. \end{aligned} \tag{3.2}$$

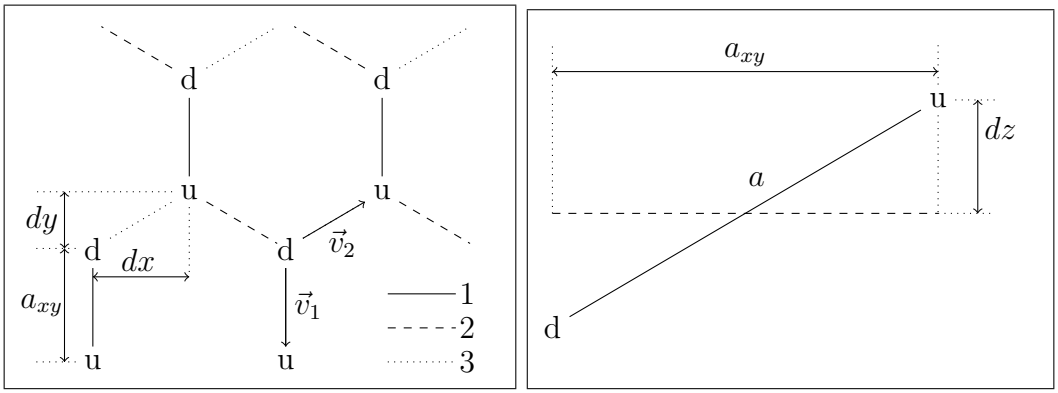


Figure 3.1: Distances necessary for the construction of the ice I_h lattice, in the x,y plane with the bond labels (left), and in the y, z plane (right).

Inserting (3.1) into the last equation yields

$$\begin{aligned} -\frac{3}{2}a_{xy}^2 &= a^2(\cos(109.47°) - 1) \\ a_{xy} &= a\sqrt{\frac{2}{3}\left(1 - \cos(109.47°)\right)}. \end{aligned} \tag{3.3}$$

With $\cos(109.47°) = -0.33331 \approx -1/3$ the variables can be written in terms of a:

$$\begin{aligned} a_{xy} &= \frac{\sqrt{8}}{3}a \\ dz &= \frac{1}{6}a \\ dy &= \frac{\sqrt{2}}{3}a \\ dx &= \frac{\sqrt{6}}{3}a \end{aligned} \tag{3.4}$$

The construction of the periodic oxygen structure can be achieved by duplicating a *"worm"*, defined along the y-axis. As can be seen in Fig. 3.1 and even better in Fig. 2.4 (left) on page 19, a periodically appearing pattern is a worm starting at the lower left corner with an up position, followed by the next three positions in y-direction (u,d,u,d). For the next layer the pattern is simply inverted with respect to the displacement along the z-axis (u $\leftrightarrow$ d). This pattern is periodic in the x, y direction, while the layer configurations repeat itself every second layer.

Nearest neighbor labeling

The nearest neighbors of a given molecule are identified with the bond at which they are attached. That way any two closest neighbors, connected by a bond, will identify each other with the same label, the label of their shared bond. In order to achieve this the following convention is introduced: The neighbor in z direction (along the c-axis) gets the label 0, while the neighbor that is merely in y direction (with respect to the hexagonal plane projection) is identified as 1-neighbor. In Fig. 3.1 the 0-bond is not shown and the 1-bond is represented by a solid line. The remaining neighbors are now labeled counterclockwise from the 1-neighbor. In the figure the 2-bond is depicted as a dotted line, while the 3-bond is shown as a dashed line.

3.2 Spherical layer system

For the energy minimization the hexagonal lattice structure with periodic boundary conditions has some disadvantages. By far the biggest one is the enforced box size, since the system can neither expand nor contract. A structure with open boundary conditions is needed, while the resulting boundary effects should not influence the structural analysis. Due to the long-range Coulomb interactions only a small subset of elements at safe distance to the boundaries may be considered not influenced by boundary conditions. It is thus necessary to minimize the boundary surface.

A layer-based spherical configuration is of advantage, since it provides a small surface area and the elements at the center of the sphere have approximately the same distance to the boundary in every direction. This configuration is constructed by starting with a central element and adding all its nearest neighbors. Every subsequent layer is added by attaching all nearest neighbors of the previous layer to the set of molecules. In order to fulfill the ice rules, which are easily implemented in the cubic case (see Sec. 3.1), the spherical molecule system can be constructed by first building a cubic system with side length of at least the diameter of the spherical system. A central molecule can be chosen as origin and layers are added, using the following *breadth-first search* [32]:
Starting with the origin molecule, all nearest neighbors are appended to a list. Beginning at the top of the list, the molecules are added to the system and their nearest neighbors are attached to the end of the list again. This is done until the wanted depth (in number of layers) is reached. Thus each layer is completely added one after another in contrast to the depth-first algorithm, where each branch is followed to the deepest level, adding all molecules along the path to the system.

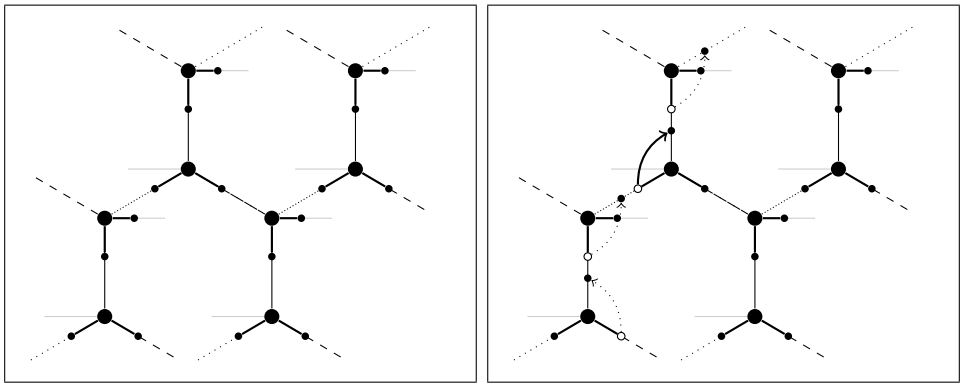


Figure 3.2: (left) Ordered ice I_h hydrogen arrangement projected to the x,y plane. (right) Scheme of a molecule flip and corresponding corrections. Gray lines indicate the bonds in the z direction.

3.3 Arrangement of the hydrogen atoms

As mentioned in Sec. 2.2.2, there exist many configurations, satisfying the ice rules.

Ordered arrangement

One possibility to arrange the hydrogen atoms periodically in the hexagonal oxygen lattice is achieved by reducing the number of hydrogen configurations to two principle ones. The first configuration places the hydrogen atoms on the c-axis (bond 0) and along the y-direction (bond 1), while the other configuration places the hydrogen atoms along the remaining bonds in x-y direction (bonds 2, 3). These configurations are then assigned to the up respectively down displacements of a given layer (see Fig. 3.2 (left)). This assignment is inverted for the neighboring layers in order to satisfy the ice rules. The resulting system is relatively ordered.

Random arrangement in the spherical layer system

In the spherical layer system it is possible to construct a random hydrogen arrangement by starting with the central molecule as seed. Layer after layer the hydrogen atoms are randomly placed on the neighboring bonds, such that the ice rules are satisfied. This will eventually run into dead ends, because there exist sites with three neighbors from lower layers. If all of these bonds are already occupied, the algorithm has to go back to a previous, still valid, layer and continue until at last a valid configuration is found. Two drawbacks appear, for one this is only possible for open systems and furthermore the required computation time explodes with system size.

An efficient alternative starts with the previously discussed ordered arrangement. Random molecules are then *flipped*, meaning that two random neighboring bonds are chosen on which the hydrogen atoms are placed. The resulting violations of the ice rules are corrected by a flip of the involved neighbors, causing new violations and so on (see Fig. 3.2 (right)). The processing of the neighbors is breadth-first, treating all directions simultaneously and therefore allowing in principle closed loops. In general the corrections will process until they reach an open end, which restricts an efficient application to open systems. A sufficient number of flips is needed in order to reach true random arrangements, at least proportional to the number of molecules. Similar approaches can be seen in [33, 29].

4 Numerical Methods

4.1 Energy minimization

In order to be able to simulate water with the help of a given model, it is important to have essential knowledge about the behavior of the model. Monte Carlo simulations (explained in detail below) in the ice phase present difficulties due to the low temperatures. Thus, when sampling the phase space it is necessary to know the principal structure of the model in this phase. In the best case the model assumes the same arrangement known from experiments. Therefore the ideal hexagonal ice I_h structure is a good initial configuration. Following the dynamics of the system it is then possible to minimize the energy, thus finding a local minimum describing the properties of the model.

4.1.1 Equations of motion

The motion of a molecule can be separated into translation of the center of mass and a rotation of the rigid body (see Sec. 2.1.2). Thus each molecules equation of motion is obtained with respect to the position of the center of mass $\vec{r}_{cm}$ and the rotation angle φ.

One equation of motion relates the magnitude of the torque T to the rotation angle. The torque may also be defined as the rate of change of the body's angular momentum $\vec{L}$

$$\vec{T} = \frac{d\vec{L}}{dt}. \tag{4.1}$$

For a fixed rotation axis, which is applied during any given time step, the angular momentum is proportional to the angular velocity $\vec{\omega}$: $\vec{L} = I\vec{\omega}$. I is the moment of inertia, defined for a body of a finite number (N) of mass points as

$$I = \sum_{i=1}^{N} m_i r_i^2. \tag{4.2}$$

Here r_i are the shortest distances between the mass points and the rotation axis. Considering that the rotation axis is fixed during any given time step, it is sufficient to consider the scalar lengths only, relating the strength of the torque to the

rotation angle about the axis

$$T = I\frac{d\omega}{dt} = I\frac{d^2\varphi}{dt^2}. \tag{4.3}$$

It is important to notice that the moment of inertia is directly dependent on the orientation of the torque and hence has to be calculated in each time step.

The second equation of motion relates the center of mass force $\vec{F}_{cm}$ to the center of mass position (see (2.3)) via the total mass of the rigid body M. Thus the following differential equations are the foundation of the dynamics used in the minimization, where T_{cm} is the torque about the center of mass.

$$\begin{aligned} M\ddot{\vec{r}}_{cm} &= \vec{F}_{cm} \\ I\ddot{\varphi} &= T_{cm} \end{aligned} \tag{4.4}$$

4.1.2 Energy minimizing time integration

It is important to notice that the energy minimization in this context does not aim for the global minimum. On the contrary the local minima are of great interest in order to classify the mean structure of the ice I_h phase.

In a naive approach, the energy minimization can be realized by a simple time integration omitting the particles velocities, neglecting the particles inertia. This is achieved by setting the state after each time step to be the new initial configuration with zero velocity. That way the particles only move in direction of the acting, decreasing forces, directly towards a local energy minimum. This can be imagined as the motion of the state in the phase space, being a potential energy landscape (see Fig. 4.1). Since the system experiences no inertia, the state cannot move beyond this local point as long as the time step is small enough. Integration of the equations of motion (4.4), with zero velocity and a time step Δt, yields the following change of variables:

$$\begin{aligned} \Delta r_{cm} &= \frac{1}{2}\frac{\vec{F}_{cm}}{M}(\Delta t)^2 \\ \Delta\varphi &= \frac{1}{2}\frac{T_{cm}}{I}(\Delta t)^2 \end{aligned} \tag{4.5}$$

This method requires a sufficiently small time step, still large enough to avoid long computational times. The exactness of the final configuration depends on the convergence criterion. It seems natural to consider either the change in energy per time step as a parameter for convergence, or to choose the mean force which approaches zero at an energy minimum. In terms of computational costs the

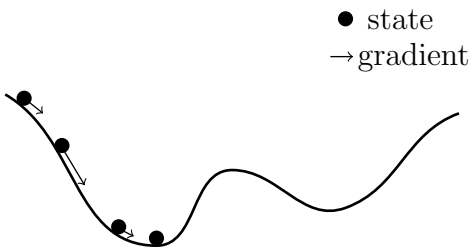


Figure 4.1: The energy minimization can be considered as a motion downhill the phase space potential landscape.

latter is more convenient, because in this case the calculation of the energy may be omitted.

4.2 Monte Carlo simulation

Markov chain Monte Carlo simulations are suited to solve statistical problems numerically, which might be hard or even impossible to solve analytically. In physical literature they are often simply referred to as *Monte Carlo (MC)* simulations, which in general includes simple sampling as well. The basis is a Markov process, generating state $n + 1$ stochastically from state n without having any knowledge of the previous states. This stands in contrast to random sampling, where each state is generated independently of each other with a given probability. Thus a MC algorithm has to make sure that the desired probability distribution is asymptotically obtained for large n, which may be achieved by importance sampling, as discussed below.

When using MC simulations some problems may arise. Due to the direct dependence on the previous state, the subsequent configurations become correlated, which has to be considered in the error analysis. Furthermore in the case of continuous systems, more error sources arise, for example rounding errors, resolution restrictions of the random number generator and histogram discretization errors.

An introduction to Markov chain Monte Carlo simulations for discrete and continuous systems is given in [34].

4.2.1 Canonical ensemble

The canonical ensemble, sometimes also referred to as Gibbs or Boltzmann ensemble, is a set of states of the system, each being a possible state in which the real system may reside. These states are Boltzmann distributed, which means that the

probability for a given state i is

$$p_i = \frac{1}{Z} e^{-\beta E_i}, \tag{4.6}$$

where E_i is the energy of the state, $\beta = (kT)^{-1}$ the inverse temperature and Z the *partition function*. The partition function is defined as the sum of Boltzmann factors over all possible states $\{i\}$

$$Z = \sum_{\{i\}} e^{-\beta E_i}. \tag{4.7}$$

The canonical ensemble is suitable for calculations of observables at a fixed temperature, where the energy is allowed to fluctuate. For a discrete system with N possible states the expectation value of an observable O is given by

$$\langle O \rangle = \frac{1}{Z} \sum_{i=1}^{N} O_i e^{-\beta E_i}. \tag{4.8}$$

4.2.2 Importance sampling

The idea of importance sampling is to generate states according to their probability. This way states that contribute less to the expectation value (4.8) are not generated as often as those that contribute more.

For practically any interesting system the number of possible states diverges. In case the configurations are generated with their *Boltzmann weight* $e^{-\beta E_i}$, the expectation value (4.8) is calculated as the arithmetic average over the number of measurements N

$$\langle O \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} O_i. \tag{4.9}$$

Since the number of measurements is always finite, it is only possible to obtain the *estimator of the expectation value*

$$\bar{O} = \frac{1}{N} \sum_{i=1}^{N} O_i. \tag{4.10}$$

It is important to point out that normally it is not possible to generate Boltzmann distributed configurations. This is the reason for the implementation of Markov chains, where a new configuration is generated from an existing one. As a result a time series of configurations is obtained.

Assume the system is in a configuration k, out of which it shall transit to configuration l with probability $W^{lk} = W[k \to l]$. The set of all transition probabilities

defines the *transition matrix*, characterizing the Markov process,

$$W = \left(W^{lk}\right) \tag{4.11}$$

with the following properties:

1. Ergodicity:
$p_k > 0 \quad \wedge \quad p_l > 0 \Longrightarrow \exists n > 0$ s.t. $(W^n)^{lk} > 0$
i.e. every point in phase space is reachable

2. Normalization:
$\sum_l W^{lk} = 1$

3. Balance:
$\sum_k W^{lk} p_k = p_l$

3'. Detailed balance:
$W^{lk} p_k = W^{kl} p_l$

where 3' is a stronger condition than 3. It reproduces the property of balance, when the sum over l is taken and property 2 is used.

The resulting time series of Boltzmann weighted configurations replaces the canonical ensemble average with a time average over the Markov chain, as long as the corresponding transfer matrix satisfies the above conditions.

4.2.3 Metropolis algorithm

The condition of detailed balance can be satisfied by a number of transition matrices. A common choice is the *Metropolis algorithm* [35]. It only requires to calculate the energy of the configuration, providing a general and simple algorithm.

Consider the system to be in a configuration k and by some update a configuration l is proposed. The new configuration is accepted with the probability w^{lk}, determined by the ratio of the Boltzmann probabilities (4.6). This ratio may be larger than one, in which case the probability is set to 1

$$w^{lk} = \min\left[1, \frac{p_l}{p_k}\right] = \begin{cases} 1 & \text{for } E_l < E_k \\ e^{-\beta(E_l - E_k)} & \text{for } E_l > E_k \end{cases}. \tag{4.12}$$

If the proposed configuration is rejected the previous configuration k is kept and the process is started over again. An *acceptance rate* can be defined by the ratio of accepted proposals over total proposals.

Probably the most crucial point is to choose appropriate update techniques to propose a new configuration. This new configuration is proposed with a certain

probability $f(l,k)$ normalized such that $\sum_l f(l,k) = 1$, depending on the update procedure. Thus the transition probability for $l \neq k$ is given by

$$W^{lk} = f(l,k) w^{lk}, \tag{4.13}$$

The requirement of detailed balance leads to

$$\frac{W^{lk}}{W^{kl}} = \frac{f(l,k)}{f(k,l)} e^{-\beta(E_l - E_k)} \stackrel{!}{=} e^{-\beta(E_l - E_k)}. \tag{4.14}$$

This results in the condition

$$f(l,k) = f(k,l). \tag{4.15}$$

It is important that the updates used in a Metropolis MC simulation ensure the detailed balance and ergodicity ($f(l,k) > 0$) criterion, which sometimes can be less obvious.

4.2.4 Updates

The discussion above shows the importance of appropriate updates, generating subsequent Markov chain elements. Below two suitable update techniques are presented and one update method that violates detailed balance.

In praxis the Metropolis acceptance step is always the same. A given molecule's configuration is saved and afterwards the molecule is updated. The new energy is calculated, which in the case of long-range interaction may cost some computation time, and related to the old molecules energy. According to the Metropolis probability the new configuration is then accepted or denied, while in the latter case the given molecule's change is reversed by returning to the saved configuration.

Rotation update

The rotation update, as the name suggests, rotates a random molecule. It is realized by generating a random rotation axis in spherical coordinates by drawing a random azimuthal and polar angle. This way the rotation axis is equally distributed in spherical coordinates. Furthermore a random angle out of a given update range is drawn by which the molecule is rotated.

The update requires at least four random numbers and the probability to propose a rotation from configuration k to l is the same as vice versa, satisfying detailed balance.

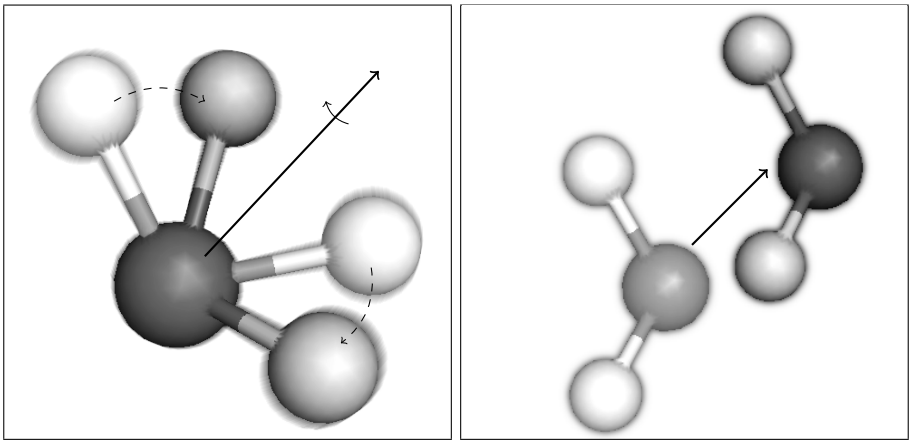


Figure 4.2: Possible Monte Carlo updates include the rotation about an arbitrary axis (left) and translation in an arbitrary direction (right).

Translation update

Again the name of the update reveals the function, the objective is to move a random molecule in space. To this extent a random direction, equally distributed on a unit sphere, is generated and a translation length is determined randomly. The translation length is restricted to a chosen range in order to improve the acceptance rate by omitting unlikely new configurations.

Thus the update requires at least four random numbers. Detailed balance is fulfilled since the probability to propose a translation in direction $\vec{a}$ is the same as proposing a translation along $-\vec{a}$.

When simulating water in and near the ice phase, the previously explained ice I_h structure is very unstable. In order to preserve the structure, the translation update needs to be restricted further. Considering the indistinguishability of the water molecules, a possible solution would be the restriction of each water molecule to a confined region about its initial position, assuming a molecule traveling far away from its position is replaced by another. Thus the position of a single molecules "oscillates" about its minimum position, requiring a precise knowledge of the structure. This is possible in the canonical ensemble, where volume and particle number, and thus the density, are constant. As indicated above, this update is not suitable for Monte Carlo simulations in the liquid phase but rather in the near ice and ice phase.

Rotation update depending on the torque

In order to obtain better acceptance rates the idea comes to mind to propose a rotation update about the axis in direction of the torque (see Sec. 2.1.2). This way the molecule will move in the direction it would tend to naturally, which is not

random. However, this violates detailed balance. Once the molecule is rotated the acting forces change and the previous configuration can possibly not be proposed again. Biasing may be a strategy to preserve detailed balance [36], in which case the acceptance probability additionally depends on the proposal densities.

4.3 Error estimation

Error estimation is an important part of any statistical analysis, since it reveals the quality of the results [37]. Often used as an estimation is the simple arithmetic mean, for example in the case of importance sampling MC simulations. The exact expectation value $\langle O \rangle$ is thereby approximated by the expression

$$\overline{O} = \frac{1}{N} \sum_{i=0}^{N} O_i, \tag{4.16}$$

where O_i are the individual measurements. The inexactness of this estimation can be expressed by the variance

$$\epsilon_{\overline{O}}^2 = \sigma_{\overline{O}}^2 = \left\langle \left(\overline{O} - \langle \overline{O} \rangle \right)^2 \right\rangle = \langle \overline{O}^2 \rangle - \langle \overline{O} \rangle^2. \tag{4.17}$$

Inserting (4.16) into this expression, considering linearity of the expectation value, yields

$$\sigma_{\overline{O}}^2 = \frac{1}{N^2} \sum_{i,j=1}^{N} \langle O_i O_j \rangle - \frac{1}{N^2} \sum_{i,j=1}^{N} \langle O_i \rangle \langle O_j \rangle. \tag{4.18}$$

This can be sorted into terms with the same and those with different indices

$$\sigma_{\overline{O}}^2 = \frac{1}{N^2} \sum_{i=1}^{N} \left(\langle O_i^2 \rangle - \langle O_i \rangle^2 \right) + \frac{1}{N^2} \sum_{i \neq j}^{N} \left(\langle O_i O_j \rangle - \langle O_i \rangle \langle O_j \rangle \right). \tag{4.19}$$

The first term is proportional to the variance of the individual measurements. The second term is symmetric in O_i and O_j, which can be used to rewrite the sum into $2 \sum_{i=0}^{N} \sum_{j=i+1}$. Furthermore the time translation invariance of the measurements results in an invariant expectation value $\langle O_i \rangle = \langle O_{i+n} \rangle$, as well as an invariant correlation $\langle O_i O_j \rangle = \langle O_{i+n} O_{j+n} \rangle$ for any $n \in \mathbb{N}$. Collecting with respect to $k = |i - j|$ and applying the above arguments finally yields

$$\sigma_{\overline{O}}^2 = \frac{1}{N} \left[\sigma_{O_i}^2 + 2 \sum_{k=1}^{N} \left(\langle O_1 O_{1+k} \rangle - \langle O_1 \rangle \langle O_{1+k} \rangle \right) \left(1 - \frac{k}{N} \right) \right], \tag{4.20}$$

where the additional factor appears since for any k exactly $N-k$ correlations are possible.

In case of N uncorrelated measurements the second term in (4.20) vanishes and the variance of the mean value reduces to $\sigma^2_{\overline{O}} = \frac{1}{N}\sigma^2_{O_i}$.

4.3.1 Integrated autocorrelation time

For correlated measurements, the variance of the arithmetic mean (4.20) strongly depends on the second term. It can be written as

$$\sigma^2_{\overline{O}} = \frac{\sigma^2_{O_i}}{N} 2\tau'_{O,int}, \tag{4.21}$$

by introducing the *(proper) integrated autocorrelation time* $\tau'_{O,int}$

$$\tau'_{O,int} = \frac{1}{2} + \sum_{k=1}^{N} A(k)\left(1 - \frac{k}{N}\right), \tag{4.22}$$

as the sum over the normalized autocorrelation function ($A(0) = 1$)

$$A(k) = \frac{\langle O_i O_{i+k}\rangle - \langle O_i\rangle\langle O_{i+k}\rangle}{\langle O_i^2\rangle - \langle O_i\rangle^2}. \tag{4.23}$$

For sufficiently large k the correlation vanishes and as a consequence the autocorrelation function approaches zero. Thus, for (meaningful) simulations with $N \gg k$, the correction term in (4.22) can be safely neglected [37]. The resulting *integrated autocorrelation time* is then

$$\tau_{O,int} = \frac{1}{2} + \sum_{k=1}^{N} A(k). \tag{4.24}$$

From (4.21) it is evident that the statistical error is increased by the square root of the integrated autocorrelation time

$$\epsilon_{\overline{O}} = \sqrt{\sigma^2_{\overline{O}}} \propto \frac{\sqrt{2\tau_{O,int}}}{\sqrt{N}}. \tag{4.25}$$

This emphasizes the importance of large N with respect to the integrated autocorrelation time in order to reduce the statistical error. Moreover, serious simulations should provide an estimate of the autocorrelation times.

Furthermore, the expectation value of observables derived from measured observables gets *biased* for large integrated autocorrelation times. One example is the specific heat $C = \beta^2 V(\langle e^2\rangle - \langle e\rangle^2)$, where e is the mean energy and V the

volume, with the following expectation value (see [37])

$$\langle \hat{C} \rangle = C \left(1 - \frac{2\tau_{e,int}}{N} \right). \tag{4.26}$$

Thus, for large correlations the heat capacity is underestimated. This effect also exists in the uncorrelated case, when the integrated autocorrelation time of the uncorrelated case is defined as $\tau_{e,int} = \frac{1}{2}$. Usually the bias-corrected estimator is therefore introduced

$$\hat{C}_{corr} = \frac{N}{N-1} \hat{C}. \tag{4.27}$$

Numerical estimation

Ironically, the parameter ensuring the quality of a statistical estimation has to be estimated itself. When estimating the autocorrelation function, the expectation values in (4.23) are replaced with mean values. Remembering the time translation invariance, the unnormalized estimator of the (biased) autocorrelation function can be obtained as

$$\hat{A}'(k) = \overline{O_i O_{i+k}} - \overline{O_i} \cdot \overline{O_{i+k}} = \overline{(O_i - \overline{O_i})(O_{i+k} - \overline{O_i})}, \tag{4.28}$$

where $\hat{A}'(0)$ is the normalization factor. As discussed previously, the correlation is supposed to vanish with increasing k, resulting in a "running estimator" of the integrated autocorrelation time

$$\hat{\tau}_{O,int}(k_{max}) = \frac{1}{2} + \sum_{k=1}^{k_{max}} \frac{\hat{A}'(k)}{\hat{A}'(0)}. \tag{4.29}$$

With large k the running estimator approaches $\tau_{O,int}$, but at the same time the statistical error increases as less data points are available. Experience has shown that it is reasonable to self-consistently compute the running estimator until $k \geq 6\hat{\tau}_{O,int}(k_{max})$.

The obtained information allows an estimation of the quality of the results. It is of great interest to find MC updates with low autocorrelation times. If this is not possible, short MC simulations may be performed in advance in order to adjust the number of sweeps in between measurements, such that the correlations between measurements minimize.

4.3.2 Binning and Jackknife error estimation

In order to reduce the effects of possible correlations in the error estimation, it is possible to consider only the mean values calculated from subsets of measurements.

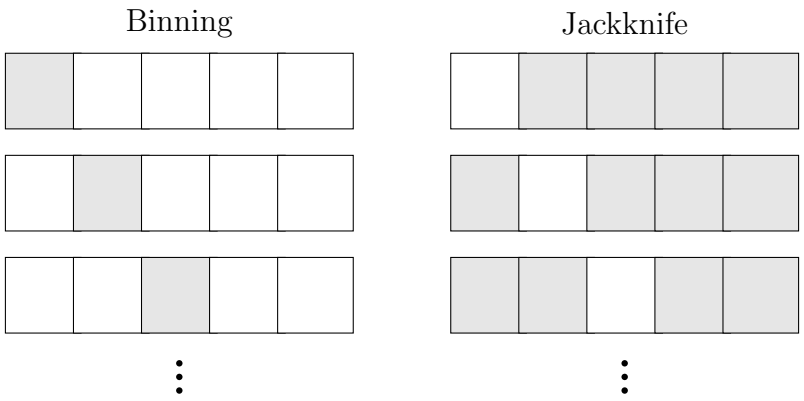


Figure 4.3: Scheme of the binning (left) and jackknife (right) blocking.

Consider N more or less correlated measurements O_i and divide this set into N_B subsets of size k (see Fig. 4.3 (left)), such that $N = N_B k$. The mean values $O_{B,n}$ of the subsets are uncorrelated, if the blocks are chosen large enough ($k \gg \tau$). Then the statistical error can be obtained from the variance of the mean values σ_B^2 with the bias-corrected estimator

$$\epsilon_{\overline{O}}^2 = \sigma_B^2 / N_B = \frac{1}{N_B(N_B-1)} \sum_{n=1}^{N_B} \left(O_{B,n} - \overline{O}\right)^2 . \tag{4.30}$$

This approach is called the *binning method*.

The *Jackknife* method reduces the bias problem for non-linear combinations of basic variables. Consider N_B large, overlapping "Jackknife" blocks, each containing all data except for one binning block, i.e. $N-k$ elements (see Fig. 4.3 (right)). The mean value of such a Jackknife block $O_{J,n}$ can thus be computed by

$$O_{J,n} = \frac{N\overline{O} - kO_{B,n}}{N-k}. \tag{4.31}$$

These mean values are trivially correlated since the same data enters $N_B - 1$ different blocks and consequently the variance will be much smaller than in the binning case. For example in the case of $N_B = N$ the variance of $O_{J,n}$ is very small, as all the mean values are practically the same. This correlation can be corrected by multiplying the variance with $(N_B - 1)^2$, yielding

$$\epsilon_{\overline{O}}^2 = \frac{N_B - 1}{N_B} \sum_{n=1}^{N_B} \left(O_{J,n} - \overline{O}_J\right)^2 , \tag{4.32}$$

where $\overline{O}_J$ is the mean value of the $O_{J,n}$.

5 Optimization

5.1 Parallelization with graphics processing units

The use of *general purpose graphics processing units* (GPGPU) in science has recently been of large interest. A variety of different applications have been investigated. For many applications large speed up factors could be gained. Considered among others are molecular dynamics [38, 39, 40], fluid dynamics [41] as well as general boundary value problems [42] to name only a few. Furthermore, the combination of GPUs to a cluster [43] provides large performance improvement at relatively low cost.

The dynamic minimization technique, used in this work, is well suited for parallelization with *graphics processing units* (GPUs) using *OpenGL*, as all molecules are handled in the same manner at the same time. In this case, it is not necessary to manage the GPU or the data transfer within the graphics card. In principle, graphics cards are composed of several so-called shading units, which access the same memory and can be parallelized by the driver. This can be used by rewriting the core functions of the simulation in a way the driver understands. Before this can be discussed, some basics need to be explained.

5.1.1 Basic working principle of GPUs

The main purpose graphics cards are designed for, is to generate output to a display. This includes the translation of 3D images, for example from a simulation, into 2D images represented by pixels. Because the original purpose is the presentation of images visible to the human eye, the processes in the GPU are designed for single precision floating-point calculations. The advantage of graphics cards lies in their large number of processing units (shading units), allowing the original purpose of 3D rendering operations, image processing, etc., performing the same computations on a large volume of data.

On a GPU the data is stored in *textures*, which can be understood as 2D-arrays of 4-component vectors (see Fig. 5.2). They were originally designed to store the color channels *RGBA*.

A general graphics task on a GPU involves several steps; it runs through the graphics hardware *pipeline*, see Fig. 5.1. At specific stages in the pipeline small programs, so called shaders, are executed on the shading units and perform the

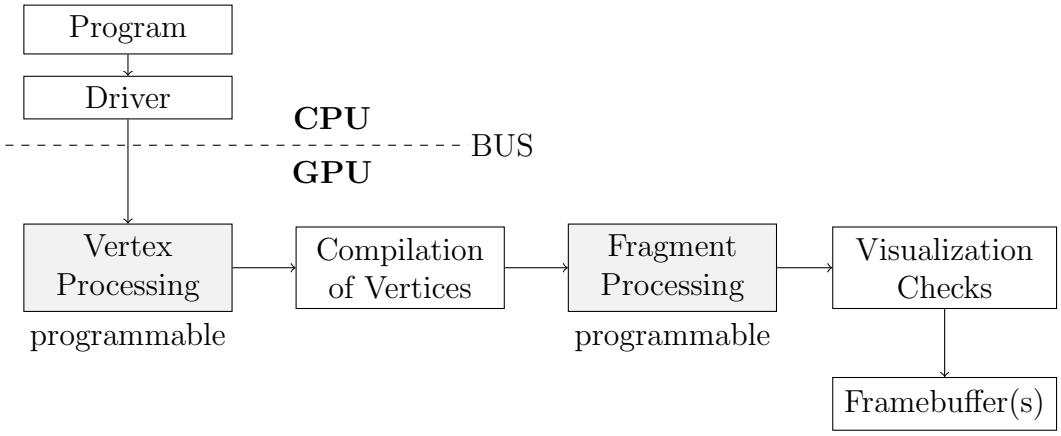


Figure 5.1: Simplified illustration of the GPU pipeline.

necessary work. The driver translates the program code, written for example in OpenGL, into commands that the GPU Front End can understand and sends all necessary data to the GPU. This data transfer between the RAM and the GPU is the largest bottleneck in many GPU computations. On the GPU the 3D input is then transformed into a 2D representation by interpreting a 3D scene from a certain position and reducing it to the vertices visible in a 2D picture. This is done by the programmable *vertex shader*. Afterwards these vertices are grouped into new primitives and converted to a set of fragments in screen space, which is referred to as compilation of vertices in the figure. In these *fragments* the state information of the screen pixels are stored. The *pixel* or *fragment shader* then assigns a color to each pixel, which is based on pixel values from one or more textures. This may include mathematical operations, in order to determine the right color by interpolations, etc. At last, the GPU checks the output's depth and alpha channel to determine whether a certain pixel in the frame buffer is updated or not. In the end, the result is send to the framebuffer, from where it can be displayed.

5.1.2 Parallelization of the minimization algorithm with OpenGL

The GPU pipeline may be used for parallelized processes, where the same operation is performed on every point of the system. This is the case for molecular dynamics simulations of particle systems, or the minimization technique used in this work, as all particles are updated simultaneously, requiring the whole information of the previous system state. MC simulations including single particle updates are more problematic, because the updates need to be performed consecutively.

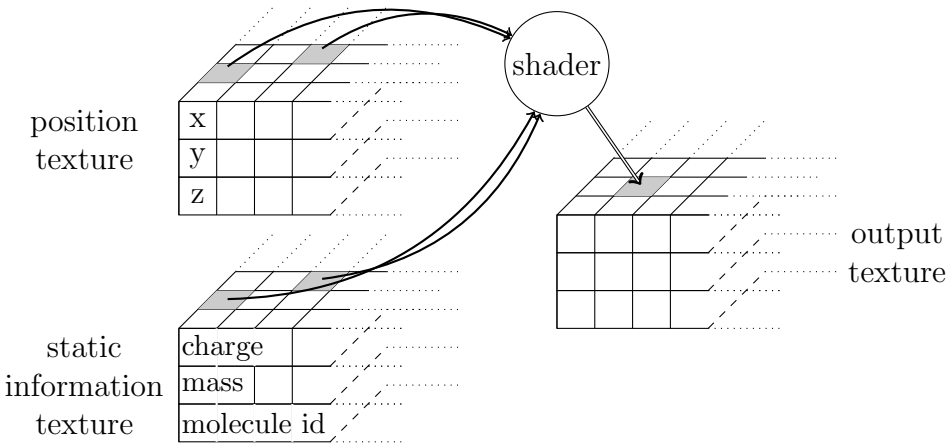


Figure 5.2: With the help of shaders, data stored on textures can be processed and directed to an output texture.

The energy minimization is parallelized in the following way: First, the molecule system is initialized by the CPU and stored into the RAM, according to the previously described geometry and algorithms (see Chap. 3). The data of the system is stored in one-dimensional arrays and then uploaded from the RAM to the GPU into the corresponding 2D textures. The textures are only composed of 3-component vectors, because the important information can be reduced to the 3D position of the atoms (→*position texture*) and the charge, mass and molecule identification number (→*static information texture*).

For each operational step a separate shader (composed of vertex and fragment shader) is executed, directing its output into a single texture, see Fig. 5.2. In each case, the vertex shader obtains the texture coordinates that need to be updated. As for the minimization the whole texture needs to be updated, the shader draws an imaginary, filled rectangle over the whole virtual screen. This way the GPU gets the information that the fragment operations need to be applied on every texture element. At this level the optimized parallelization takes place, since the GPU driver is developed to autonomously distribute the tasks onto the shading units as efficient as possible. The fragment shader thus continues in parallel with the actual physical computation. Each minimization step requires the current gradient of the energy. Thus, one shader calculates the force acting on every atom via a loop over all other texture elements, except for the ones belonging to the same molecule. The force is then stored in an additional texture. With the help of the calculated forces the integration step is performed by the next shader, calculating the center of mass forces, the moment of inertia, and the torque before translating and rotating the molecules. The result is stored in a texture that is eventually swapped with the position texture. Yet another shader calculates the energy in the same manner as the force and stores it in a further texture.

Parameter	Time (10^3s)	Factor
AMD Opteron Dual-Core 2.6GHz (single)	74.664	1
Intel core2Quad 2.4GHz (single)	62.053	1.2
ATI Radeon 4870	0.978	76

Table 5.1: Performance comparison of GPU/CPU via 5000 minimization steps ($\Delta t = 0.2$fs) with 4713 molecules (17 layers).

The data transfer within the graphics card is very fast. On the other hand, the transfer between the RAM and the graphics card is very slow. Thus, it is necessary that the measurements are not performed in every step, but with an appropriate step interval. In case of the minimization method, this means that the mean center of mass force of the system, or alternatively the mean energy, is calculated after prefixed intervals. Otherwise, the performance drops drastically as too much time is spend on the slow data transfer. For system sizes of interest an interval of at least 20 time steps is required. Then the time spent on the data transfer from the GPU is below the order of the computational time spent on the interval of minimization steps. This effect can be observed for small systems, where the GPU cannot show its real power because the actual computation is very fast but the data transfer happens too often.

The parallel simulation on the GPU may be compared to the non-parallelized CPU simulations performed so far. These were executed on the working station at hand, an Intel core2Quad 2.4GHz, and the ITP Leipzig computation cluster *grawp* equipped with AMD Opteron Dual-Core processors. Since OpenGL is, in principle, independent of the graphics cards manufacturer, the OpenGL program was tested on an Nvidia Tesla[1] and an ATI Radeon 4870 card. For a system of 4713 molecules the minimization routine was executed on both GPUs and CPUs and compared (see Tab.5.1). The improvement factor has to be considered carefully. For one the CPUs and GPUs were not produced in the same years and in addition the GPUs did perform the operations with single precision, whereas the CPU computations were performed with double precision. The latter point is only of importance, if double precision is required, in which case the GPU performance drops strongly. Simulations with single precision on CPUs, on the other hand, do not significantly improve the performance. In Fig. 5.3 the somewhat unfair comparison of a single CPU core and the ATI graphics card (with 800 shading units) is shown, in order to demonstrate the power of GPU simulations. The computational time is plotted versus the system size in number of molecules. Above a size of about 5000 molecules the simulation times on the CPU become very large,

[1]The Nvidia card actually presented some unresolved difficulty for larger systems, which might be due to unwanted optimizations of the Nvidia driver. Because of this it does not appear in the performance comparison.

as long as no cutoff is introduced. The computational time on the graphics card increases much slower. This is due to the fact that on a CPU each molecule is considered one at a time, performing a loop over all other molecules, while on the GPU many molecules can be processed in parallel from the same shared memory. Thus, simulations using a GPGPU are a good and generally cheap alternative to small CPU clusters, especially considering their weak increase of simulation time with system size. In the end, it is important to notice that graphics cards produce a lot of heat, which has to be taken into account for a suited location.

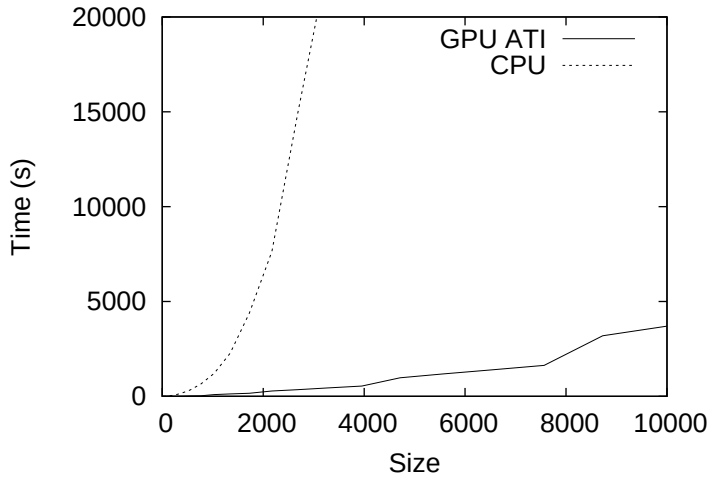


Figure 5.3: Computation time dependence on system size of CPU and GPU.

5.2 Multihit Metropolis algorithm

One of the limitations in the Monte Carlo simulation of a water system in the ice I_h configuration is the rapidly increasing integrated autocorrelation time τ_{int} with decreasing temperature. As described in Sec. 4.3.1, large integrated autocorrelation times lead to an increased statistical error as well as a bias in the estimation of observables. The problem lies in the low acceptance rate at low temperatures. It cannot be avoided by choosing a smaller update range, because this again would results in larger correlations between subsequent states. Increasing the number of sweeps in between measurements would reduce the correlation, but would drastically increases the computation time. Therefore a solution that reduces the integrated autocorrelation time, while increasing the simulation time mildly, is of great interest.

One solution is the *multihit* Metropolis algorithm [34]. The idea is to consider a given Metropolis update multiple (n) times in order to provide the possibility for any molecule to reach lower energy states. Updating each molecule n times offers

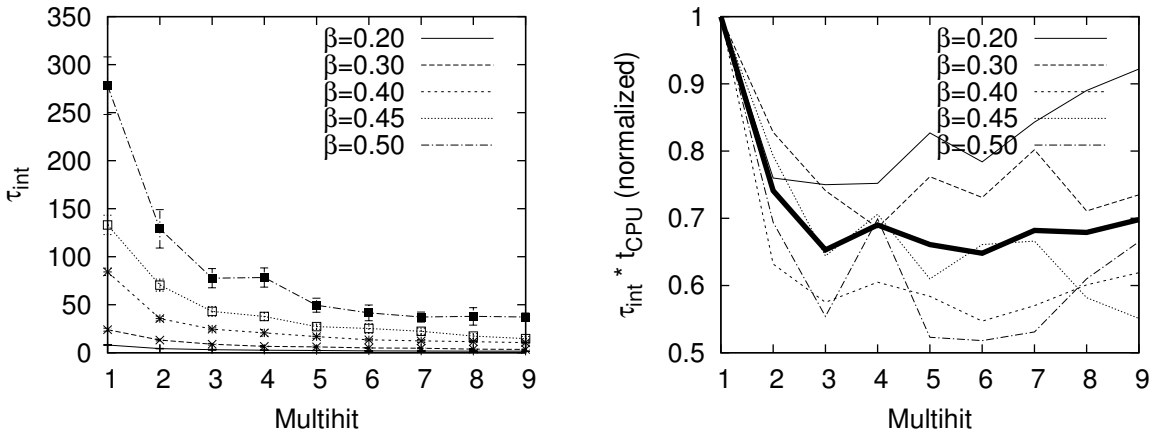


Figure 5.4: τ_{int} (left) and $\tau_{int} \cdot t_{CPU}$ (right) versus number of multihits for different inverse temperatures β. The mean product of τ_{int} and simulation time is shown as thick line. The relative errors for the right plot are omitted for clarity.

the advantage that each resulting energy can be passed into the multihit routine as the old energy again. That way the most expensive part of the update has been reduced by almost a factor of two, as in each multihit routine there are now $n+1$, instead of $2n$, calculations of the energy.

The multihit Metropolis algorithm was tested for a system of 128 particles with cutoff 4Å at different inverse temperatures. It can be seen in Fig. 5.4 (left) that the integrated autocorrelation time decreases with larger multihit number approaching a limit specific to the temperature. At the same time, the acceptance rate increases along with the simulation time. The optimal multihit number n is determined by a low integrated autocorrelation time and a low simulation time. It is thus obtained at the minimum of the product of both quantities. In Fig. 5.4 (right) this product is plotted versus the multihit number. The relative errors from the integrated autocorrelation time are of the order of 10%, which explains the fluctuations in the plot. For inverse temperatures below 0.50 and nearest neighbor interaction, a multihit $n = 6$ is optimal producing small integrated autocorrelation times at reasonable computation time.

This method proves very effective in reducing the integrated autocorrelation time. The appropriate multihit number depends strongly on the temperature range and the energy calculation method. For the given system a performance factor of almost two was achieved at low temperatures with the multihit Metropolis algorithm.

6 Numerical Results

In this chapter, the numerical results are presented. The study of the structure of Tip4p parameterizations in the ice I_h configuration begins with the investigation of the interaction between two water molecules with their minimal energy configuration. Afterwards, random spherical ice systems are considered with the help of the minimization technique introduced in Sec. 4.1.
In the end, the influence of the results on Monte Carlo simulations is investigated and discussed.

6.1 Local energy minimum of the Tip4p water model in the ice I_h phase

In order to understand the energy minimum of the solid ice phase, it is helpful to understand the basic interaction between two isolated water molecules first.

6.1.1 Interaction between two water molecules

Consider an initial configuration with two molecules A and B. Due to symmetry reasons, a minimal energy configuration requires both molecular planes to be perpendicular (see Fig. 6.1). In addition, the dipole of B lies along the connecting line of the two oxygen atoms. This way the hydrogen atoms of B are at the same distance from each hydrogen atom of A. As a consequence, the components of the

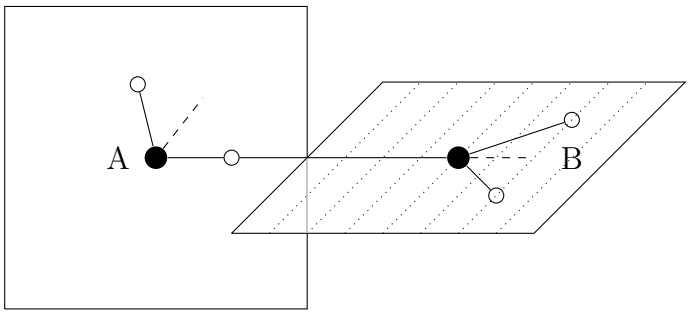


Figure 6.1: Perpendicular molecular planes allow a symmetric arrangement of the hydrogen atoms, resulting in a minimal energy contribution of the hydrogen atoms.

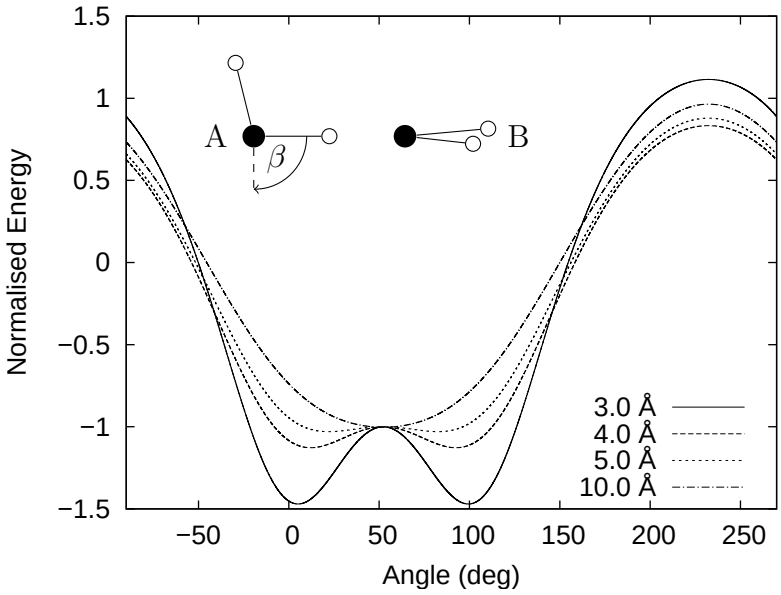


Figure 6.2: Angular energy distributions for two water molecules (Tip4p), with perpendicular molecular planes, at different distances. The distribution is normalized such that the energy at 52.27° is -1.

acting forces, perpendicular to the molecular plane of A, cancel. B shall be fixed for the moment. The three rotational degrees of freedom of the variable molecule A reduce to one. For this degree of freedom, the angular energy distribution is obtained by small rotations of A, measuring the interaction energy at each step (see Fig. 6.2). For large distances the energy minimum is reached when the dipole of A is parallel to the dipole of B, while an anti-parallel dipole arrangement maximizes the energy. Reducing the distance between A and B results in two local minima, which approach the angles 0° and 104.52° with decreasing distance.

The symmetry breaking (see Fig. 6.3) begins at d_0 and is caused by the Coulomb interaction, since the Lennard Jones interaction (only between oxygen atoms) is independent of the angle. The critical distance depends on the model and parameterization used. It is almost identical for the Tip4p/2005 and Tip4p/Ice parameterization, while it deviates largely in the case of the Tip3p model. Nonetheless, the qualitative behavior is the same for all models considered.

It can be argued that for large distances the Coulomb interactions of the hydrogen atoms are almost equal, such that a symmetric configuration becomes favorable. At short distances, however, the attractive interaction between a hydrogen charge and a foreign oxygen charge is larger than the repulsion from the foreign hydrogen charges. This is due to the $1/x$ behavior of the Coulomb energy, resulting

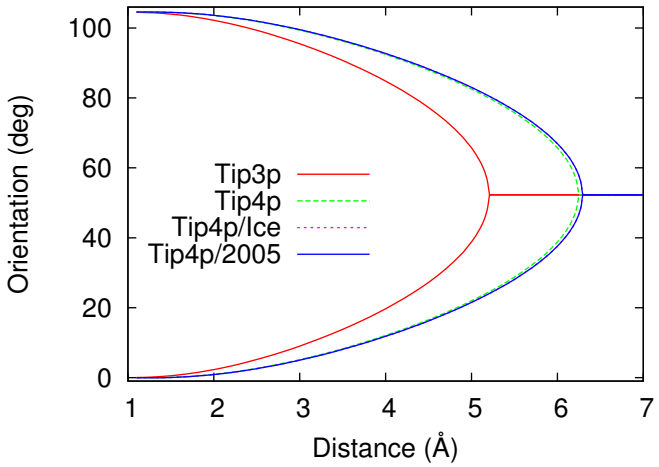


Figure 6.3: Minimal energy angle corresponding to Fig. 6.2 versus distance between the oxygen atoms. At d_0 a single minimum (symmetric configuration) splits into two minima (h-bond configuration).

in a larger contribution of a given Δx at small distances. The resulting minimum configurations at close distance are those, where one of both hydrogen atoms of A are directed towards the oxygen atom of B, forming the well known *hydrogen bond* (h-bond).

In the field of dynamical systems, such an effect is called bifurcation, where a small smooth change in a parameter (in this case the distance) causes a sudden change in behavior (in this case the orientation of the molecule). The single minimum splits into two perfectly symmetric minima at the bifurcation point d_0. This is called *pitchfork bifurcation* [44]. Bifurcation theory predicts that the behavior near d_0 is proportional to $(d_0 - d)^\alpha$. For the different parameterizations the following exponents were obtained near the bifurcation point with a nonlinear fit:

$$\begin{aligned} \alpha_{Tip3p} &= 0.496 \pm 0.001 \\ \alpha_{Tip4p} &= 0.495 \pm 0.001 \\ \alpha_{Tip4p2005} &= 0.497 \pm 0.001 \\ \alpha_{Tip4pIce} &= 0.497 \pm 0.001 \end{aligned} \tag{6.1}$$

This indicates a general square-root-like behavior ($\alpha \approx 0.5$) near d_0 for all parameterizations considered.

The bifurcation points, as result of the same fit, on the other hand differ:

$$\begin{aligned} d_{0,Tip3p} &= (5.202 \pm 0.001)\text{Å} \\ d_{0,Tip4p} &= (6.242 \pm 0.001)\text{Å} \\ d_{0,Tip4p2005} &= (6.290 \pm 0.001)\text{Å} \\ d_{0,Tip4pIce} &= (6.290 \pm 0.001)\text{Å} \end{aligned} \tag{6.2}$$

This distance is larger than the distance between neighboring molecules in and near the ice phase, which is of order 3Å [25]. Especially the next nearest neighbors, but also the nearest neighbors, will influence each molecule's orientation depending on the distance, thus resulting in possibly different lattice constants for different models. It is clearly visible that there exist significant differences between the Tip3p and Tip4p model.

Consider now the favorable hydrogen bond pair configuration with molecule B variable and molecule A fixed. Again, two rotational degrees of freedom can be eliminated, such that the forces acting on the hydrogen atoms of B are equal. This results in a lower net torque and consequently a lower interaction energy. The angular dependence of the interaction energy on the remaining degree of freedom is shown in Fig. 6.4, where the distance between A and B is minimized for each angle. For the Tip4p parameterizations the following angles were found to minimize the energy:

$$\begin{aligned} \beta_{Tip3p} &= -(27.2 \pm 0.01)^\circ \\ \beta_{Tip4p} &= -(34.7 \pm 0.01)^\circ \\ \beta_{Tip4p2005} &= -(34.5 \pm 0.02)^\circ \\ \beta_{Tip4pIce} &= -(32.9 \pm 0.02)^\circ \end{aligned} \tag{6.3}$$

It is interesting to notice that these configurations are similar to only one of the possible water molecule arrangements, within the boat configuration (see Sec. 2.2.1) of the ideal ice I_h (also called c-trans configuration [29]). The corresponding angle may be calculated as $\beta = \arctan(\frac{dy}{2dz}) = -54.74^\circ$ (see (3.4)). For these energetically favorable configurations the oxygen-oxygen distances d_i take the following values:

$$\begin{aligned} d_{Tip3p} &= (2.745 \pm 0.001)\text{Å} \\ d_{Tip4p} &= (2.701 \pm 0.001)\text{Å} \\ d_{Tip4p2005} &= (2.721 \pm 0.001)\text{Å} \\ d_{Tip4pIce} &= (2.767 \pm 0.001)\text{Å} \end{aligned} \tag{6.4}$$

This already shows that the structure of the minimal ice I_h configuration will

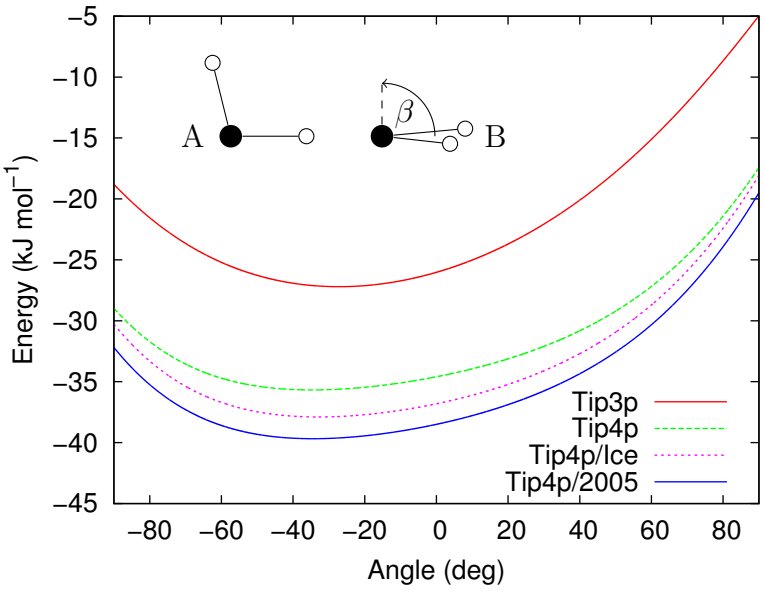


Figure 6.4: Energy distribution for two water molecules in the favorable h-bond pair configuration. The distance was chosen for each angle such that the energy is minimized.

probably depend on the parameterization used. Also, the specific energy levels and minima vary with the parameterization.

6.1.2 Local energy minimum in an ice I_h configuration

The minimizations were performed with a time step of $\Delta t = 0.2$ fs, in order to provide only small spatial displacements and to keep the system stable. This time step is smaller than in usual molecular dynamics simulations, for example concerned with the melting point of ice I_h, where often the time steps are of order 1fs (for example [45]). With the given order of force parameters the initial spatial update range per time step is of the order 10^{-6}Å, sufficiently small for the simple time integration method. As criterion for convergence of the minimization the mean center of mass force was required to be reduced to 1/100 of the standard initial values for the given time step.

The initial configurations of the minimizations were spherical ice I_h structures as described in Sec. 3.1 with lattice constant $a = 2.764$Å [2]. In order to obtain representative results, 100 different realizations of the ice rules (Sec. 2.2.1) were constructed by a random hydrogen arrangement (Sec. 3.3). For the investigation of unit cell distances and angles, only up to 12 molecules from the central core of the system were considered in order to avoid boundary effects as much as possible.

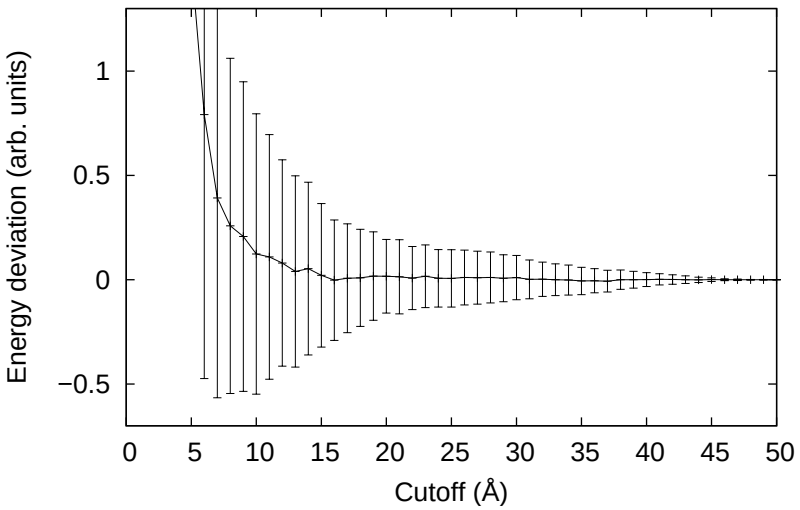


Figure 6.5: Mean deviation (of set with $r <$ cutoff) from the limiting energy ($r =$ cutoff) and the corresponding sample standard deviation.

Long range interaction

The Coulomb interaction is effective on a large range. This can be seen in Fig. 6.5, where the energy contribution per cutoff sphere is shown. For a set of 100 different spherical systems in a random ice I_h configuration (with 9997 molecules), the initial contribution to the energy of the central molecule from within a sphere of a given cutoff radius was calculated. All molecules were restricted to their initial configurations. Because different random configurations have slightly different energies, the energy with cutoff at 50Å was taken as reference. Then the deviations from the reference energy were averaged. The error-bars in Fig. 6.5 are the sample standard deviations. This shows that the approach of nearest neighbor interaction ($r <$ 3Å) is not suitable for the simulation of water molecules. For acceptable energies in the minimization, the cutoff should be chosen to be at least 25Å. The findings in this work correspond to previous discussions about the long-range interactions in molecular dynamic simulations of water models [46, 47, 48].Some of the previous works suggest that the use of cutoffs may not allow for a correct description of mesoscopic and macroscopic properties.

Size Scaling

The minimization was performed for a set of 100 random configurations at different sizes. On the CPU, all-molecule interactions were only possible up to a spherical system with 17 layers (4713 molecules). For larger simulations a cutoff of 25Å was introduced, which was tested to provide the same results in the range of the

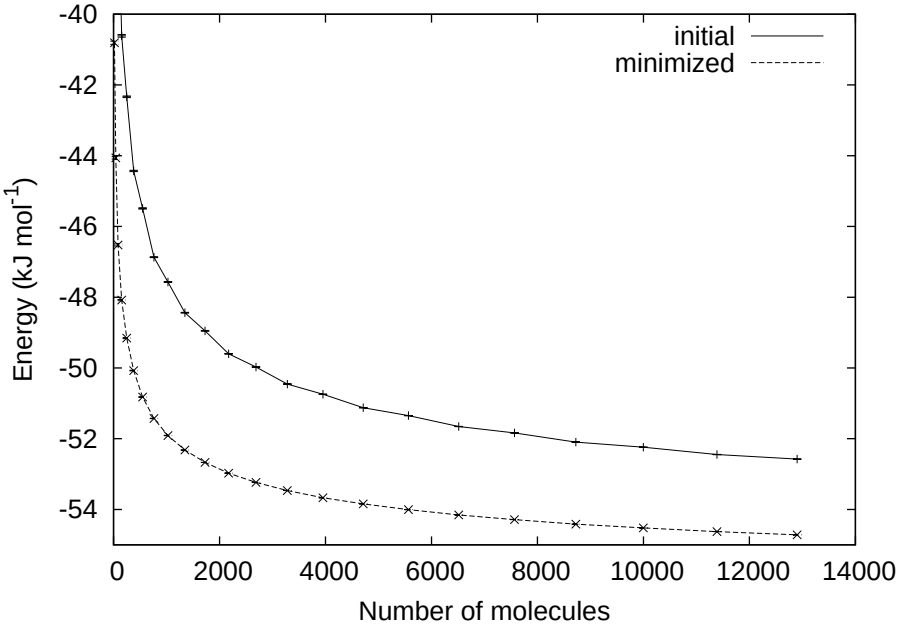


Figure 6.6: Mean energy per molecule of the Tip4p parameterization before and after minimization depending on system size in number of molecules.

statistical error and is therefore suitable for the minimization. In Fig. 6.6 the scaling behavior of the mean energy per molecule is presented. With increasing system size the mean energy decreases and approaches an asymptotic value for the initial as well as the minimized mean energy. Interpolating the available data with a power law, the minimized mean energy for the infinite system was estimated to be $-(57.3 \pm 0.1)^{\mathrm{kJ}}/_{\mathrm{mol}}$ for the Tip4p model.

The strong dependence on the system size is clearly due to the long range effect of the interaction potentials, especially the Coulomb potential. As can be seen in Fig. 6.5, the interaction energy of a single molecule increases with larger cutoff, and consequently the calculated potential energy is overestimated for systems smaller than the range of the interaction. Furthermore, for larger systems, the boundary effects on the mean energy are suppressed, such that the mean value approaches the value for a system of infinite size.

Also, the mean angles and distances in the unit cell vary with size, though for large systems the deviations are smaller. The best estimation obtained in this work is given later.

Minimal violation of ground state energy degeneration

Throughout this study the energy degeneration of ground states, described in Sec. 2.2.2 and responsible for the residual entropy, was found to be not exact but

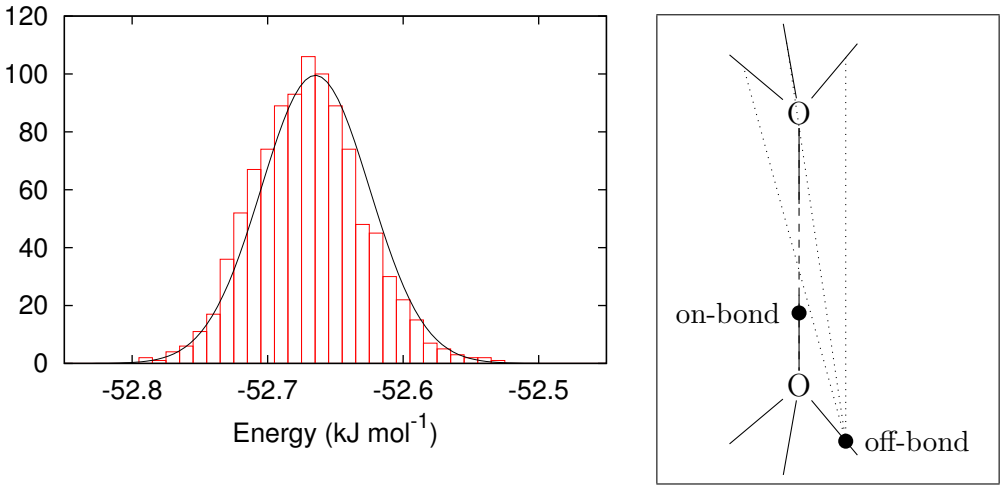


Figure 6.7: (left) Gauss like distribution of local energy minima (1726 molecules) with original Tip4p parameters. (right) For off-bond hydrogen atoms the distances to interacting off-bond hydrogen atoms may differ.

to fluctuate slightly (see Fig. 6.7 (left)). This histogram with an energy width of $0.01^{\mathrm{kJ}}/_{\mathrm{mol}}$, was obtained by minimization of 1000 clusters with 1726 Tip4p water molecules each. Defining the i-th moment about the mean $\mu_i = \frac{1}{n}\sum_{j=1}^{n}(x_j - \overline{x})^i$ allows to classify the distribution of measurements, where $\mu_2 = \sigma^2$ is the variance. The skewness $g_1 = \frac{\mu_3}{\sigma^3}$ and the kurtosis $g_2 = \frac{\mu_4}{\sigma^4} - 3$ have to vanish in case of a normal distribution. In fact the distribution of energies of local minima may be approximated by a gaussian distribution ($g_2 = 0.16$) with a small positive skew ($g_1 = 0.19$), such that the mass is shifted slightly to the left. A finite-size dependence is not evident, also due to less data for larger systems.

It may be said that the energy degeneration is lifted already for the Tip4p model. As a consequence, there does exist a global minimum (see Fig. 6.7 (left)). This is not due to quantum effects, but is caused by the model itself. The original argument for energy degeneration assumes the hydrogen atoms in the central point of each oxygen bond, which is not fulfilled in the present models. Consider the pair interaction between two molecules with one distinct bond of interest (see Fig. 6.7 (right)). The energy contribution from the hydrogen-hydrogen interaction with one hydrogen on-bond and the foreign counterpart off-bond remains symmetric, because the distances are identical by definition of the tetrahedral structure. In case both hydrogen atoms are off-bond the possible interaction distances differ. This is enhanced by the fact that the water angle differs from the tetrahedral angle. Thus, the total energy depends on the individual orientation of the molecules. Considering long-range interactions this effect increases. For example, the energy of a molecule also depends on the the orientations of the next-nearest neighbors.

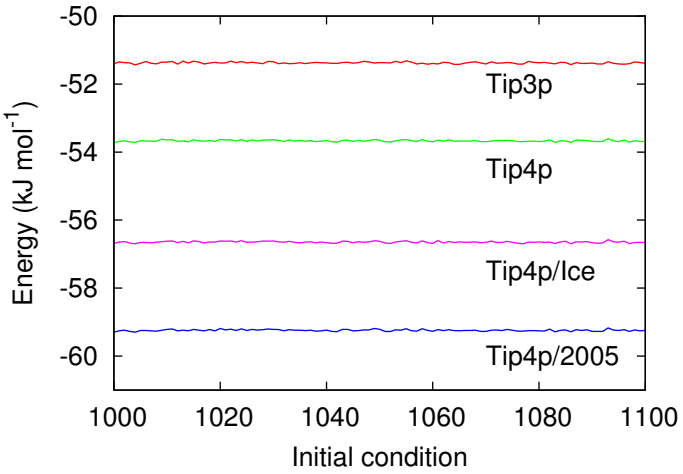


Figure 6.8: The specific minimized ground state energy deviates slightly for different initial conditions (3953 molecules), satisfying the ice rules.

Thus the total interaction potential of a molecule is not isotropic, leading to small differences in energy for different random hydrogen arrangements. This is observable for all parameterizations, plotting the minimal energies of 100 random samples with 3953 molecules each (see Fig. 6.8), satisfying the ice rules. The different energy levels are consistent with the two molecule minimal energy levels (see Fig. 6.4). While the mean energy is clearly depending on the specific parameterization, the deviations from the mean are consistently less than 1% of each mean value.

In nature, deviations are measured between the local minima and the global minimum of ice XI. The differences, however, are so small that they can only be experimentally distinguished at low temperatures [49]. Thus, it is possible in practice to freeze the orientation of a water system below 100K in a local minimum, which is reproduced by the present zero temperature minimizations, preventing the achievement of the global minimum [31].

The present data, which is only a very small subset of ground states, shows that the degeneration of energy ground states is already lifted for all parameterizations at hand, but within a range of less than 1%. These small deviations, however, are only of importance at very low temperatures, where the system might actually freeze in one particular ground state, which is not necessarily the global minimum. Thus, for the subset of ground states considered, this rather supports the assumption of degenerate ground states. Furthermore, this can be interpreted as a hint that the considered parameterizations of the Tip4p water model present a residual entropy in the continuous space.

Unit cell structure

One important basis for the discussion of the unit cell of ice I_h is the assumption of a regular structure. In the simulations the water models exhibit stable local minimum configurations. In order to obtain qualitative results about the mean ice I_h structure, the results of 100 random configurations satisfying the ice rules, were averaged. The elements of a unit cell show the same parameterizations within their statistical errors. In the literature the unit cell elements of ice are characterized by four principle parameters. For one, there are the distances to the neighbors in the hexagonal plane a_h and the distance to the nearest neighbor in z-direction (along the c-axis) a_c. In addition, there are the angles between the neighbor along the c-axis and the neighbors in the hexagonal plane θ_c as well as the angles between neighbors within the hexagonal plane θ_h. A scheme of a tetrahedral unit cell element is shown in Fig. 6.9, and results of the minimization of different parameterizations are presented in Tab. 6.1.

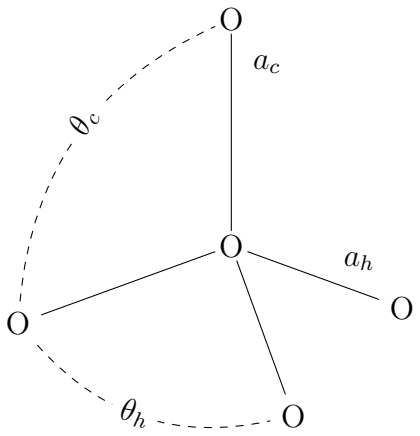


Figure 6.9: Structure parameters of the hexagonal unit cell elements

	Tip3p		Tip4p		Tip4p/2005		Tip4p/Ice	
a_c (Å)	2.728(2)	0.016	2.713(1)	0.003	2.734(1)	0.003	2.776(1)	0.003
a_h (Å)	2.726(2)	0.016	2.712(1)	0.003	2.733(1)	0.003	2.775(1)	0.003
θ_c (deg)	109.4(5)	5.0	109.5(3)	2.7	109.5(3)	2.7	109.4(3)	2.6
θ_h (deg)	109.3(5)	4.9	109.4(3)	2.7	109.4(3)	2.6	109.4(3)	2.6

Table 6.1: Structure parameters of the hexagonal unit cell elements (see Fig. 6.9) as result of the energy minimization and their standard deviations.

The elements of the unit cell are arranged with angles around 109.4° within the hexagonal plane and from the c-axis. These mean angles (see Tab. 6.1) are close

to the tetrahedral angle 109.47° but show variances of order 2%. Thus, the unit cell elements are arranged in principle tetrahedral with small deviations. For all parameterizations the distance to the nearest neighbor along the c-axis is larger than within the hexagonal plane, but the differences remain within the standard deviation. The mean lattice constants, on the other hand, show distinct differences for different parameterizations but are of the same order as the energy minimum distances obtained for the two body energy minimum (see (6.4)). For the Tip3p model the variances are larger than for the rest of the Tip4p parameterizations. This suggests that the Tip3p model presents a less regular ice structure in comparison to the considered Tip4p parameterizations.

The general results can be compared with the data of Röttger et al. [25], who investigated the dependence of the lattice constant of H_2O on temperature with synchrotron radiation. All of the considered parameterizations, except Tip4p/Ice, present a smaller lattice constant than obtained in the experimental measurements $a_{c,exp} > 2.741$Å. Moreover, the c/b ratio (see Sec. 2.2.1) was experimentally confirmed to be smaller than the theoretical value. This was identified as a result of θ_h being larger than θ_c. This is clearly not the case for the parameterizations considered in this analysis. For example, the original Tip4p parameterization results in a ratio $(c/b)_{Tip4p} = 1.634 \gtrsim 1.633 + (c/b)_{theo}$ within statistical errors.

The discussion about the structure of the Tip4p parameterizations and the Tip3p model may be summarized as follows: The mean structure of the oxygen atoms in the ice I_h phase is shown to be in principle tetrahedral for all parameterizations. It can be observed that the angles within the hexagonal plane seem to be slightly smaller on average than the angles from the c-axis, but the values are still within the statistical error. As expected from the investigation of two molecules, the lattice constants for an energetic minimum differ for the parameterizations considered. All of the lattice constants in the hexagonal plane are within the variance of the lattice constant along the c-axis. The values are presented in Tab. 6.1.

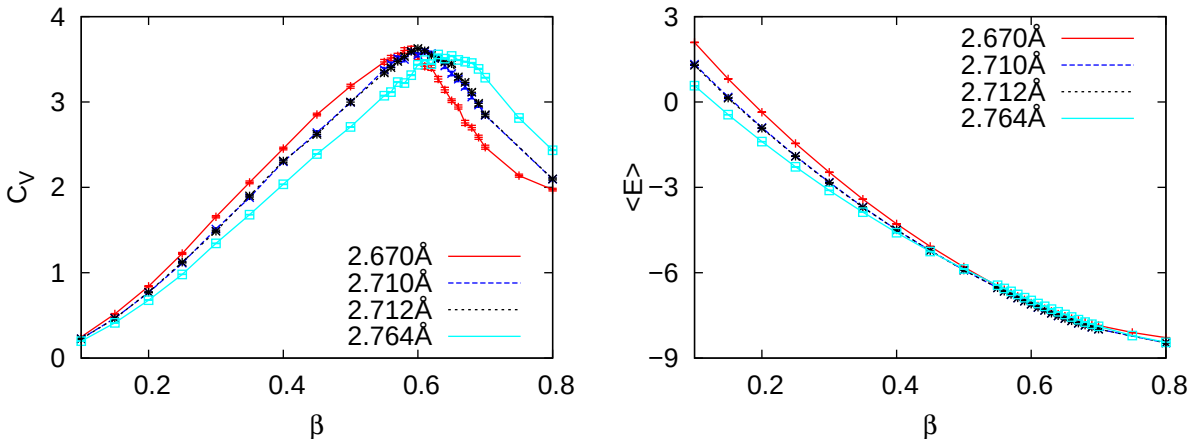


Figure 6.10: Influence of selected lattice constants on the heat capacity (left) and the mean energy (right) of 128 water molecules with fixed oxygen positions.

6.2 Monte Carlo simulations

Monte Carlo simulations are performed on a periodic hexagonal lattice for the original Tip4p parameterization only. The previous section suggests that all considered parameterizations show a similar qualitative behavior. Observables of interest are the mean energy per molecule $\langle e \rangle$ and the heat capacity $C_V = \beta^2 V(\langle e^2 \rangle - \langle e \rangle^2)$. All simulations are performed with lowest possible integrated autocorrelation times. This is achieved by first performing short simulations in order to determine the integrated autocorrelation times, which are then used as the number of sweeps between measurements. The random numbers are generated by the *Mersenne Twister* (MT19937) generator from the Boost library. For the error estimation the Jackknife method (see Sec. 4.3.2) is applied with 32 bins, each containing 1024 measurements. The energy units used in the simulations are $^{\text{kcal}}/_{\text{mol}}$, which are the units of the original parameterization (see Tab. 2.1). As a consequence, the inverse temperature is given in units of $^{\text{mol}}/_{\text{kcal}}$, and a conversion to Kelvin is possible via $T = 503.35\text{K}/\beta$.

Influence of the lattice constant

The effect of the lattice constant on MC results is investigated. In order to keep the lattice constant fixed, only rotation updates are considered. The investigated system is composed of 128 molecules with nearest neighbor interactions only. In this case the lattice constant does influence the heat capacity and the mean energy, as can be seen in Fig. 6.10, where both are plotted against the inverse temperature. Neither the heat capacity nor the mean energy differ in their basic temperature behavior for different lattice constants. It can be seen in Fig. 6.10 (right), that for

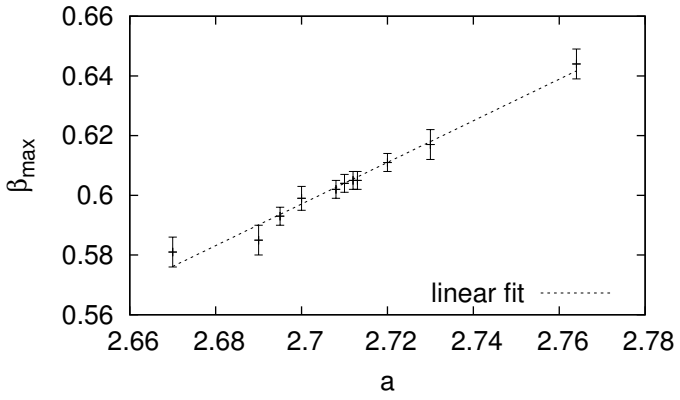


Figure 6.11: Location of the heat capacity maximum in dependence on the lattice constant of a system with 128 molecules and fixed oxygen position.

small β the mean energy increases with decreasing lattice constant. This results from the potential energy, which is inverse in the distance, because at short distances the possible differences in energies are larger. Only for larger β the minimal mean energies are obtained with the lattice constants from the minimization, even though the differences are very small. The maximal heat capacity is approximately the same for all lattice constants investigated in the interval $[2.670\text{Å}, 2.764\text{Å}]$. On the other hand, the inverse temperature β_{max}, at which the heat capacity has its maximum, increases monotonically with the lattice constant within the investigated interval (see Fig. 6.11).

Influence of the energy cutoff

In contrast to the lattice constant, the choice of the energy cutoff strongly influences the value of the mean energy and the maximum heat capacity (see Fig. 6.12). Furthermore, the inverse temperature β_{max} at which the maximum heat capacity is reached shifts for different cutoffs. It varies from $\beta_{max,4} \approx 0.61$, for the nearest neighbor interaction, to $\beta_{max,9} \approx 0.44$, for an energy cutoff at 9Å. It is important to notice that the behavior of the heat capacity and the mean energy converge for larger cutoffs. As mentioned, the simulations were performed with fixed oxygen positions, considering only rotations. In the figure it can be seen that the difference between a cutoff of 8Å and 9Å is already very small. This is consistent with usual truncation of the real part of the interaction potentials at 8.5Å in molecular dynamics and Monte Carlo simulations (see for example [45]), keeping in mind that the present simulations do not include translations. It can be said that, for MC simulations with rotations only, a minimal cutoff at 8Å is necessary for meaningful simulations.

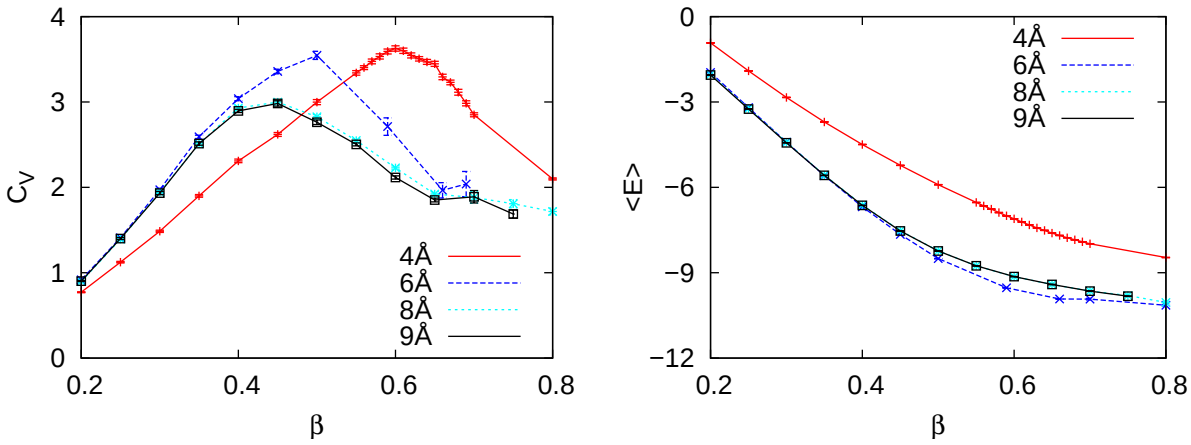


Figure 6.12: Influence of different cutoffs on the heat capacity (left) and the mean energy (right) of 128 water molecules with fixed oxygen positions.

Investigation of the translation update

The translation update is investigated in the same interval, where the rotation update presents a peak in the heat capacity. According to the previous results, the simulations are carried out with an energy cutoff at 9Å. For the translation update alone, the initialized hydrogen orientations do not change. Thus, a combination of both updates seems interesting. In this case, one of both updates is randomly selected for each Monte Carlo update. A comparison of the rotation update, the translation update, and a combination of both is shown in Fig. 6.13.

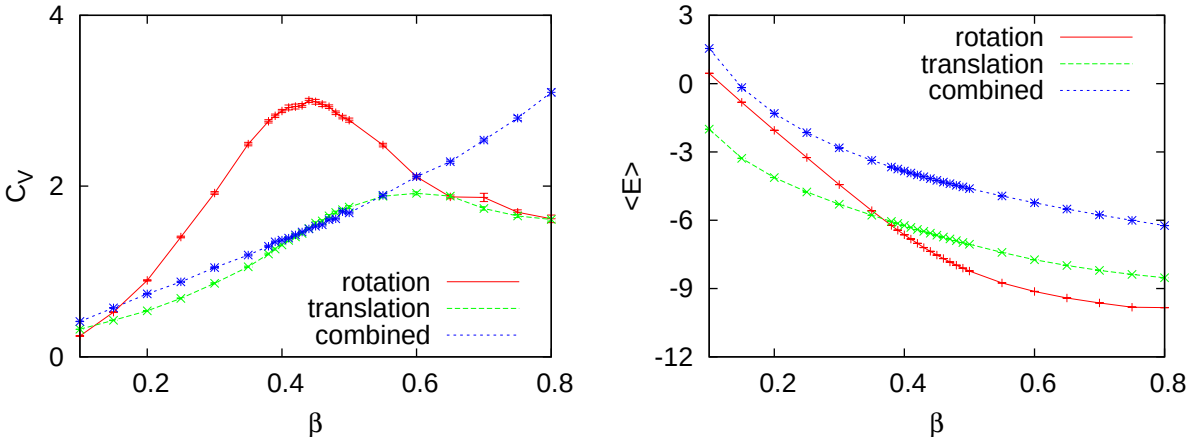


Figure 6.13: Heat capacity and mean energy from simulations with a combination of rotation and translation updates for 128 molecules with energy cutoff at 9Å.

It can be seen that the individual updates cause a peak in the heat capacity,

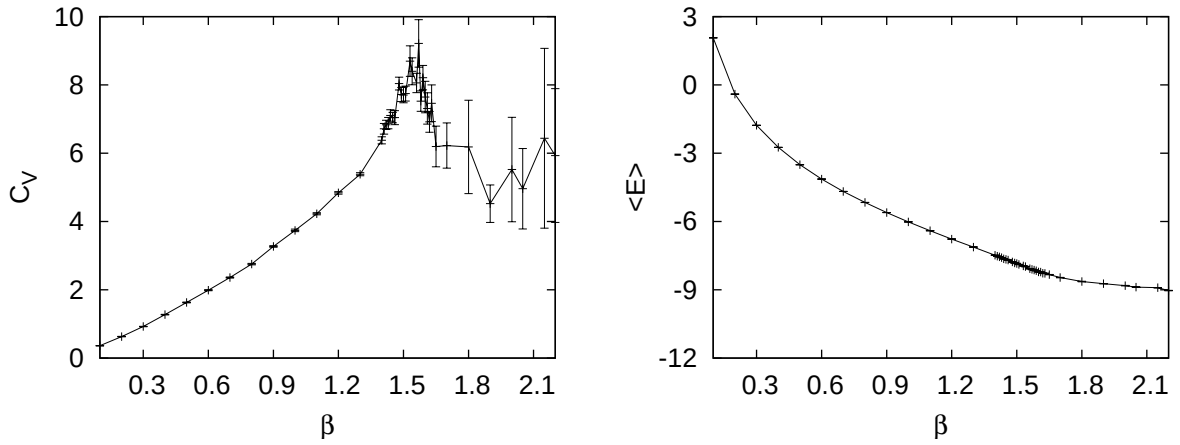


Figure 6.14: Heat capacity and mean energy from simulations with a combination of rotation and translation updates for 128 molecules with nearest neighbor interaction.

even though the peaks do not appear at the same temperature. At the beginning and at the end of the temperature interval, the heat capacities of both updates are in principle the same. A combination of both updates, on the other hand, does not show any peak but a monotonic increase of the heat capacity (Fig. 6.13 (left)). The differences are reflected in the behavior of the mean energy (Fig. 6.13 (right)). At large temperatures, the mean energy for the rotation update is larger than for the translation update but drops down fast, such that it turns out to be smaller at smaller temperatures. The mean energy of the combined update is larger than for the other cases, and decreases less with increasing β. This is understandable, as more degrees of freedom mean that the water molecules can actually carry more energy. The differences between the rotation update and the translation update can be explained by the fact that the rotation update allows the system to change between different ground states with a fixed lattice constant. The translation update, on the other hand, starts with a given ground state and a corresponding hydrogen arrangement, which cannot be changed without a rotation of the molecules. An isolated translation update is in any case of little use.

Still, the question about the monotonic increase of the heat capacity from the combined update remains. Thus, the behavior of the heat capacity and mean energy for larger inverse temperatures is of great interest. Due to large integrated autocorrelation times and consequently large computation times, the simulations are performed with nearest neighbor interactions only. Therefore, the results may only give an idea about what might happen. Fig. 6.14 shows the heat capacity and the mean energy of the combined update. For large β the correlations are very strong, but the peak in the heat capacity at about $\beta = 1.57$ is significant. Thus, the heat capacity has its peak at a much smaller temperature for the combined

update, than for either one of the single updates. It seems that the reduction to a subset of degrees of freedom changes the system qualitatively.

The results for the combined update might indicate a divergence in the heat capacity and a further investigation with finite-size scaling seems promising. In this case, new update techniques should be considered in order to handle the otherwise large correlations.

7 Conclusion

Based upon the results from the previous chapters, the following conclusions may be drawn.

The investigation of systems with two molecules showed that the interaction potentials exhibit the same properties for the Tip3p, Tip4p, Tip4p/2005 and Tip4p/Ice parameterization. At large distances the interaction is symmetric, with a single minimum configuration satisfied by aligned dipoles. When the distances are below the bifurcation distance, the energy minimum splits up into two symmetric energy minima. The resulting configurations at short distance reproduce the well known hydrogen bonds.

It was possible to show that it is not justified to assume only nearest-neighbor interactions in the presence of long-range potentials, such as the Coulomb interaction, which is dominating in the water interaction. Restricting the degrees of freedom to rotations only, there are significant differences in the qualitative results of Monte Carlo simulations, in this case the mean energy and the heat capacity, for interaction cutoffs smaller than 9Å. This is consistent with the truncation of the Coulomb and Lennard-Jones interactions at 8.5Å in previous works. If the precision of the energy is of more importance, the cutoff has to be larger.

Including long-range interactions, the simulations revealed that the Tip3p, Tip4p, Tip4p/2005, and the Tip4p/Ice water model have a stable ice I_h structure with tetrahedral oxygen arrangement. Only the Tip3p model shows relatively large deviations from a regular structure. As expected, the hydrogen atoms form hydrogen bonds with small deviations from the oxygen-oxygen bonds. None of the considered models reproduces the experimentally measured deviation of ice from the tetrahedral structure [25]. Furthermore, the obtained lattice constants as well as the energy levels of the specific parameterizations show noticeable differences. This also effects Monte Carlo results even though not as strongly as the energy cutoff. As a consequence, the maximum of the heat capacity depends linearly on the lattice constant within the considered interval. Therefore, before simulating in the ice phase, it is important to know the lattice constant of the employed model. As a rather technical remark, it can be said that graphics cards are a powerful option for deterministic simulations, such as the minimization technique. They are restricted to single precision, but allow many parallel computations at the same time with shared memory.

Throughout the minimization, it was found that the energy minima of the selected ground states are varying within a range of $< 1\%$. The energy degenera-

tion is lifted as a consequence of the Tip4p models with long-range interactions. Still, the energy of ground states is degenerated to some extend in the continuous space, allowing the ice to cross between ground states. This basic disorder implies a residual entropy. Because of the lifting of the degeneracy, however, the thermal fluctuations may become smaller than these energy differences. In this case the entropy would become lower than the one calculated in Sec. 2.2.2, which deserves further investigations [50].

Including translations heavily influences the behavior of the heat capacity and the mean energy. In this case, the heat capacity seems to have a peak at a much lower temperature, indicating a possible phase transition between the liquid and the solid phase. A finite-size analysis of this behavior is a promising approach in the discussion about phase transitions of the water model. Also a finite-size investigation of the rotational degree of freedom on a hexagonal lattice with the obtained minimal energy lattice constant and appropriate energy cutoff remains open. In addition, a variable density would be of great interest for further investigations.

In the end, water continues to be a fascinating molecule with many still unresolved questions. Even though the Tip4p models do not reproduce every property of water, they seem to be a good choice for further research in and near the ordinary ice phase.

Appendix

Selected Program Code

In the following, selected program code is presented and briefly explained. The code is reduced to the essential elements, omitting visualization, etc. The program is organized in classes (atom, molecule, system, ...), including their own functions. For example, the energy, force and torque are calculated on the atomic level and merely processed in the molecule class and higher hierarchies.

Program Code 1: Periodic hexagonal lattice

```
1  /*... Initialization ... */
2
3  //deviation from hexagonal plane and distance between two planes
4  dFromPlane      = aReal/6;
5  dplanes = aReal+2*dFromPlane;
6
7  a = sqrt(aReal*aReal-4*dFromPlane*dFromPlane); //a_xy
8
9  dZ=0; dX=0; dY=0;
10 for(int k=0; k<nz; k++){         //loop over planes
11   dZ=k*dplanes;
12   select=k%2;                   //distinguish even/odd planes
13   for(int i=0; i<nx; i++){      //loop in x direction
14     dX=0+i*sqrt(3)*a;
15     for(int j=0; j<ny; j++){    //"worm"
16       if(j==0)      {dY=0; countY=0;             dZ=k*dplanes+dFromPlane-2*select*dFromPlane;}
17       if(countY==1) {dY+=a;                      dZ=dZ-2*dFromPlane+2*2*select*dFromPlane;}
18       if(countY==2) {dY+=a/2; dX+=(sqrt(3)/2)*a; dZ=dZ+2*dFromPlane-2*2*select*dFromPlane;}
19       if(countY==3) {dY+=a;                      dZ=dZ-2*dFromPlane+2*2*select*dFromPlane;}
20       if(countY==4) {dY+=a/2; dX-=(sqrt(3)/2)*a; dZ=dZ+2*dFromPlane-2*2*select*dFromPlane;}
21
22       x[i+nx*j+nx*ny*k]=dX;
23       y[i+nx*j+nx*ny*k]=dY;
24       z[i+nx*j+nx*ny*k]=dZ;
25
26       setNextNeighbors(i+nx*j+nx*ny*k); //assign nearest neighbors
27
28       countY+=1;
29       if(countY==5) countY=1;
30     }
31   }
32 }
```

The periodic hexagonal lattice is constructed one plane after another. In each layer, nx "worms" are started in y-direction that construct lattice cites slightly above and below the layer, thus forming the desired hexagonal lattice. The required constants are defined in Sec. 3.1 (aReal=a, a=a_{xy}). Afterwards, the hydrogen atoms are positioned, such that the molecule angles are centered in the proper tetrahedral angles. While the corresponding code is straight forward, it is too long to be posted here.

Program Code 2: Spherical system

```
/* ...Create a reference periodic lattice with suitable dimension larger than the sphere... */

/* ...Select a central lattice point from the reference lattice: indexOriginHexa... */

moleculeList.push_back(indexOriginHexa);          //list of indices in reference lattice
moleculeLayerList.push_back(0);                   //list of layers molecules get in

//breath-first algorithm: add molecules from reference to spherical system
runningIndex=0;
while(layer<=numLayers){
  w[runningIndex] = systemHexa->w[ moleculeList[runningIndex] ];
  //Add neighors to the list
  for(int n=0; n<4; n++){
    for(int mol=0; mol<moleculeList.size(); mol++){
      if(w[runningIndex]->nearestNeighbor[n]->indexLattice == moleculeList[mol]) bias = 1;
    }
    if(bias!=1){
      if(layer+1<=numLayers){
        moleculeList.push_back(w[runningIndex]->nearestNeighbor[n]->indexLattice);
      }
      moleculeLayerList.push_back(layer+1);
    }
    bias = 0;
  }
  runningIndex+=1;
  layer = moleculeLayerList[runningIndex];
}

//Open boundary conditions: set boundary neighbors to NULL
for(int index=0; index<moleculeList.size(); index++) w[index]->indexLattice = -1;
for(int index=0; index<moleculeList.size(); index++){
  for(int n=0; n<4; n++){
    if(w[index]->nearestNeighbor[n]->indexLattice != -1) w[index]->nearestNeighbor[n] = NULL;
  }
  for(int n=0; n<numNeighbors; n++){
    if(w[index]->neighbor[n]->indexLattice != -1) w[index]->neighbor[n] = NULL;
  }
}
for(int mol=0; mol<systemHexa->numMolecules; mol++){
  if( systemHexa->w[mol]->indexLattice != -1 ){
    delete systemHexa->w[mol];
    systemHexa->w[mol]=NULL;
  }
}
for(int index=0; index<moleculeList.size(); index++) w[index]->indexLattice = index;
numMolecules = moleculeList.size();
```

The spherical system is constructed according to Sec. 3.2 with the help of the breadth-first algorithm. To achieve this, a periodic reference system of appropriate size (smallest side length > diameter) is created first. Both reference and spherical system are lists of pointers, including only the addresses of the molecule objects. Thus, the molecules need to be stored only once and the pointers may be quickly copied and rearranged. In order to obtain open boundary conditions, all neighbors that do not belong to the spherical system are set to “NULL”. That way they are not considered in other functions.

Program Code 3: Minimization

```
/* ...Initialization... */

while(meanForce>forceConvergence){

  //Minimization Step:
  calcForcesOpen();                          //calculate forces acting on every atom

  for(int mol=0; mol<numMolecules; mol++){
    //Vectoraddition of atomic forces
    for(int coord=0; coord<3; coord++){
      w[mol]->cmForce[coord] = w[mol]->O.Force[coord]  + w[mol]->Ocharge.Force[coord]
                             + w[mol]->H1.Force[coord] + w[mol]->H2.Force[coord];
    }
    //Calculate all torques about center of mass and store in atom class
    O.calcTorque       (centerMass[0],centerMass[1],centerMass[2]);
    Ocharge.calcTorque(centerMass[0],centerMass[1],centerMass[2]);
    H1.calcTorque      (centerMass[0],centerMass[1],centerMass[2]);
    H2.calcTorque      (centerMass[0],centerMass[1],centerMass[2]);
    //Vectoraddition of atomic torques
    for(int coord=0; coord<3; coord++){
      w[mol]->Torque[coord] = w[mol]->O.Torque[coord]  + w[mol]->Ocharge.Torque[coord]
                            + w[mol]->H1.Torque[coord] + w[mol]->H2.Torque[coord];
    }
  }

  //translation of all molecules
  for(int mol=0; mol<numMolecules; mol++){
    for(int coord=0; coord<3; coord++){
      translation[coord] =  (cmForce[coord]/totalMass)*timeStep*timeStep/2;
    }
    w[mol]->translate(translation[0], translation[1], translation[2]);
  }

  //rotation of all molecules
  for(int mol=0; mol<numMolecules; mol++){       //rotation
    w[mol]->calcMomentInertia(centerMass[0], centerMass[1], centerMass[2]);
    /* ... */
    torqueValue=0.0
    for(int coord=0; coord<3; coord++) torqueValue  += Torque[coord]*Torque[coord];
    torqueValue = sqrt(torqueValue);
    angularStep = (torqueValue/momentInertia)*timeStep*timeStep/2;
    for(int coord=0; coord<3; coord++) torqueNorm[coord] = Torque[coord]/torqueValue;
    w[mol]->rotate(centerMass, torqueNorm, angularStep);
  }

  meanForce = calcMeanForce();           //Sum all center of mass forces and do mean

  /* ...Output after deltaMeasure... */
}

/* ...Postprocessing... */
```

The minimization is an Euler time-integration without any memory of the velocities, as explained in Sec. 4.1. While the mean center of mass force is above a threshold, the known Coulomb and Lennard-Jones forces acting on every atom are calculated and stored within the atom class. Afterwards the center of mass force on each molecule (w[mol]) is calculated as the sum of all belonging atomic forces and the overall torque on each molecule is calculated as the sum of the torques on the individual atoms. Now each molecule is translated and rotated according to (4.5) with the corresponding time step. In the end the mean center of mass force is calculated in order to compare to the minimization threshold and possible output is generated after “deltaMeasure” steps.

Program Code 4: Metropolis updates

```
1  void iceSystemMC::mcUpdateRotation()
2  {
3    /* ...Initialization... */
4
5    system->calcEnergy(system->w[randMol]);
6
7    //MULTIHIT
8    for(int hit=0; hit<multihit; hit++){
9      system->w[randMol]->lock();                          //Save current state
10       zCoord     = -1 + 2*rng();                        //Random rotation direction
11       phi        = rng()*2*M_PI;
12       sinTheta   = sqrt(1-zCoord*zCoord);
13       rotationAxis[0]  = sinTheta*cos(phi);
14       rotationAxis[1]  = sinTheta*sin(phi);
15       rotationAxis[2]  = zCoord;//*/
16       randAngle = -angleLimit+rng()*angleLimit*2;        //Random angle
17     system->w[randMol]->rotate( rotationPoint, rotationAxis, randAngle);
18     system->calcEnergy(system->w[randMol]);
19     mcMetropolis(randMol,&counterCheck);
20   }
21   if(counterCheck!=0){ acceptCountRotation +=1; }
22   totalCountRotation+=1;
23 }
24
25 /* ... */
26
27 void iceSystemMC::mcUpdateOscillation()
28 {
29   /* ...Initialisation... */
30
31   system->calcEnergy( system->w[randMol] );
32
33   //MULTIHIT
34   double direction[3], deviation[3], dE=0;
35   for(int hit=0; hit<multihit; hit++){
36     system->w[randMol]->lock();                          //Save current state
37       for(int coord=0; coord<3; coord++){                //random translation
38         deviation[coord] = system->w[randMol]->O.getCoord(coord) - system->w[randMol]->O.getInitialCoord(coord);
39         direction[coord] = -osciRange + 2*osciRange*rng() - deviation[coord];
40       }
41     system->w[randMol]->translate(direction[0], direction[1], direction[2]);
42     system->calcEnergy(system->w[randMol]);
43     mcMetropolis(randMol,&counterCheck);
44   }
45   if(counterCheck!=0) acceptCountOscillation += 1;
46   totalCountOscillation+=1;
47 }
48
49 / * ... */
50
51 void iceSystemMC::mcMetropolis(int molecule, int *counter)
52 {
53   double dE = system->w[molecule]->energy - system->w[molecule]->energyLock;
54   if(dE<0){
55     system->systemEnergy+=dE;
56     *counter+=1;
57   }
58   else{
59     if(rng() < exp(-beta*dE)){
60       system->systemEnergy+=dE;
61       *counter+=1;
62     }
63     else system->w[molecule]->undo();                    //return to saved state (by lock)
64   }
65 }
```

The Metropolis MC updates are described in Sec. 4.2.4. In the selected code also the multihit optimization (see Sec. 5.2) is included. It can be seen that the old energy of the molecule is calculated before the loop over the n-hits is performed. In general, updates are performed by saving the initial configuration of the molecule, then updating it and calculating its energy, and afterwards passing it to the Metropolis function. If the update gets rejected, the initial configuration is simply restored.

Parameterfiles

In order to specify the parameters in the simulations, a parameterfile class was introduced. Any variable, double or string, has to be identified with a unique name, followed by the "=" sign and the corresponding content. It has to be terminated by either a Semicolon (;), a comment (/) or a newline (\n). In addition, strings have to be surrounded by quotes ("). Below, the fundamental variables of the routines are explained. In order to direct the output, an existing file path (FilePath) has to be specified up to the final "/", in which the program will create its own directories. The energy cutoff (neighborCutoff) may be only given in integers up to now. The all-molecule interaction (neighborCutoff= -1) option is only advised for the open spherical system, where for the periodic lattice the cutoff should be smaller than one half of the shortest side length up to now.

Parameterfile 1: Tip4p

```
Molecule Parameters:    distHydrogen    =0.9572;
                        chargeHydrogen  =0.52;
                        massHydrogen    =1.008;
                        massOxygen      =15.9994;
                        angleHydrogen   =104.52;
Oxygen:                 distCharge      =0.15;
Force Parameters:       tipA=600000.0; tipC=610.0
                        kCoulomb=332.064;
```

The water molecule class needs the distance between the oxygen and the hydrogen atoms (distHydrogen), the distance between the oxygen and the negative charges (distCharge), the molecule angle (angleHydrogen), the masses (massHydrogen, massOxygen), and a charge factor (chargeHydrogen) as fraction of the electron charge. For the interaction the Lennard-Jones parameters (tipA, tipC) and the Coulomb constant (kCoulomb) are necessary. This file is the only one to be modified for different Tip4p parameterization.

Parameterfile 2: Minimization

```
Lattice distances:      cLattice=2.764;
Next Neighbors:         neighborCutoff=-1;
Crystal Lattice Layers: Layers                  = 17;
                        SystemNumberMolecules = 4317;
FilePath="../output/";
EnergyMinimisation:     deltaMeasure            = 100;
                        timeStep                = 0.2;
                        forceConvergence        = 0.01;
                        RandomHydrogenSeed      = 1000;
```

For the minimization routine the following variables need to be specified:
The lattice constant (cLattice) and the number of layers (Layers) are necessary for the construction of the spherical system. For the analysis also the number of molecules (SystemNumberMolecules) is required, which is obtained in the minimization. The actual minimization only needs the steps between measurements (deltaMeasure), the time step in fs, the threshold of the mean center of mass force (forceConvergence), and a seed for the initial random hydrogen arrangement (RandomHydrogenSeed).

Parameterfile 3: Monte Carlo Simulations

```
Lattice dimensions:      nx=4; ny=8; nz=4;
Lattice distances:       cLattice=2.712;
Ice:                     neighborCutoff=09;
FilePath="../output/";
MonteCarlo Parameters:   seed                =1000;
                         InverseTemperature=0.10;
                         Multihit            =6;
                         OscillationRange    =0.4;
                         RotAngleRange       =1;
                         Thermalisation      =1024;
                         NumberSweeps        =302;
                         NumberMeasurements=32768;
Jackknife:               JackBins            =32;
```

For the Monte Carlo simulations the following variables are necessary:
The periodic hexagonal system is constructed from the lattice constant (cLattice) and the lattice dimensions (nx, ny, nz). In this case nx has to be a multiple of 1, ny of 4 and nz of 2. The actual MC simulation needs an initial seed (seed), the inverse temperature (InverseTemperature), the number of multihits (Multihit), the number of measurements for thermalisation (Thermalisation), the number of sweeps between measurements (NumberSweeps) and the number of actual measurements (NumberMeasurements). For the analysis the number of Jackknife bins has to be specified (JackBins) and it desired that NumberMeasurements is a multiple of JackBins. The MC updates may be influenced by restricting the rotation range (RotAngleRange ≤ 1) or the range of translations about the initial oxygen position (OscillationRange) as a fraction of the lattice constant a.

Bibliography

[1] D. Eisenberg and W. Kauzmann. *The Structure and Properties of Water*. Oxford University Press, 1969. Reprinted 2006.

[2] V. F. Petrenko and R. W. Whitworth. *Physics of Ice*. Oxford University Press, 2002.

[3] J. L. Finney. The water molecules and its interactions: the interaction between theory, modelling, and experiment. *J. Mol. Liq.*, 90(1-3):303 –312, 2001.

[4] J. L. Finney. Water? What's so special about it? *Phil. Trans. R. Soc. B*, 359(1448):1145–1165, 2004.

[5] Sir W. H. Bragg. The crystal structure of ice. *Proc. Phys. Soc. London*, 34(1):98, 1921.

[6] S. W. Peterson and H. A. Levy. A single-crystal neutron diffraction study of heavy ice. *Acta Crystallogr.*, 10(1):70–76, 1957.

[7] S. J. La Placa and B. Post. Thermal expansion of ice. *Acta Crystallogr.*, 13(6):503–505, 1960.

[8] B. Guillot. A reappraisal of what we have learnt during three decades of computer simulations on water. *J. Mol. Liq.*, 101(1-3):219 – 260, 2002.

[9] J. D. Bernal and R. H. Fowler. A theory of water and ionic solution, with particular reference to hydrogen and hydroxyl ions. *J. Chem. Phys*, 1:515–548, 1933.

[10] J. A. Barker and R. O. Watts. Structure of water; a Monte Carlo calculation. *Chem. Phys. Lett.*, 3(3):144 – 145, 1969.

[11] A. Rahman and F. H. Stillinger. Molecular dynamics study of liquid water. *J. Chem. Phys.*, 55(7):3336–3359, 1971.

[12] H. J. C. Berenden, J. P. M. Postma, W. F. vanGunsteren, and J. Hermans. Interaction models for water in relation to protein hydration. *Intermol. Forces*, pages 331–342, 1981.

[13] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.*, 79(2):926–935, 1983.

[14] M. W. Mahoney and W. L. Jorgensen. A five-site model for liquid water and the reproduction of the density anomaly by rigid, nonpolarizable potential functions. *J. Chem. Phys.*, 112(20):8910–8922, 2000.

[15] J. L. Aragones, E. G. Noya, J. L. F. Abascal, and C. Vega. Properties of ices at 0 K: A test of water models. *J. Chem. Phys.*, 127(15):154518, 2007.

[16] L. Pauling. The structure and entropy of ice and of other crystals with some randomness of atomic arrangement. *J. Am. Chem. Soc.*, 57:2680, 1935.

[17] W. F. Giauque and J. W. Stout. The entropy of water and the third law of thermodynamics. the heat capacity of ice from 15 to 273 K. *J. Am. Chem. Soc.*, 58(7):1144–1150, 1936.

[18] H. F. J. Savage and J. L. Finney. Repulsive regularities of water structure in ices and crystalline hydrates. *Nature*, 322:717–720, 1986.

[19] J. L. F. Abascal and C. Vega. A general purpose model for the condensed phases of water: Tip4p/2005. *J. Chem. Phys.*, 123(23):234505, 2005.

[20] J. L. F. Abascal, E. Sanz, R. García Fernández, and C. Vega. A potential model for the study of ices and amorphous water: TIP4P/Ice. *J. Chem. Phys.*, 122(23):234511, 2005.

[21] F. Scheck. *1.Mechanik: Von den Newtonschen Gesetzen zum deterministischen Chaos.* Springer, 1999. first published 1989.

[22] J. Lekner. Energetics of hydrogen ordering in ice. *Physica B*, 252:149–159, 1998.

[23] N. W. Ashcroft and N. D. Mermin. *Festkörperphysik.* Oldenbourg Verlag München Wien, 2005. first published 1976: "Solid State Physics".

[24] K. Lonsdale. The structure of ice. *Proc. R. Soc. Lond. A*, 247:424–434, 1958.

[25] K. Röttger, A. Endriss, J. Ihringer, S. Doyle, and W. F. Kuhs. Lattice constants and thermal expansion of H_2O and D_2O ice I*h* between 10 and 265 K. *Acta Cryst.*, 50(6):644–648, 1994.

[26] National Institute of Standards and Technology (NIST). `http://physics.nist.gov/cuu/`.

[27] B. A. Berg, C. Muguruma, and Y. Okamoto. Residual entropy of ordinary ice from multicanonical simulations. *Phys. Rev. B*, 75:092202, 2007.

[28] J. F. Nagle. Lattice statistics of hydrogen bonded crystals. I. The residual entropy of ice. *J. Math. Phys.*, 7(8):1484–1491, 1966.

[29] V. Buch, P. Sandler, and J. Sadlej. Simulations of H2O solid, liquid, and clusters, with an emphasis on ferroelectric ordering transition in hexagonal ice. *J. Phys. Chem. B*, 102(44):8641–8653, 1998.

[30] Y. Tajima, T. Matsuo, and H. Suga. Phase transition in KOH-doped hexagonal ice. *Nature*, 299:810–812, 1982.

[31] S. M. Jackson, V. M. Nield, R. W. Whitworth, M. Oguro, and C. C. Wilson. Single-crystal neutron diffraction studies of the structure of ice XI. *J. Phys. Chem. B*, 101(32):6142–6145, 1997.

[32] C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.

[33] J. A. Hayward and J. R. Reimers. Unit cells for the simulation of hexagonal ice. *J. Chem. Phys.*, 106(4):1518–1529, 1997.

[34] B. A. Berg. *Markov Chain Monte Carlo Simulations and Their Statistical Analysis*. World Scientific Publishing, 2004. reprinted 2006.

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

[36] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[37] W. Janke. Statistical analysis of simulations: data correlations and error estimation. *NIC Series*, 10:423 – 445, 2002.

[38] J. Yang, Y. Wang, and Y. Chen. GPU accelerated molecular dynamics simulation of thermal conductivities. *J. Comput. Phys.*, 221(2):799 – 804, 2007.

[39] W. Liu, B. Schmidt, G. Voss, and W. Müller-Wittig. Accelerating molecular dynamics simulations using graphics processing units with CUDA. *Comput. Phys. Commun.*, 179(9):634 – 641, 2008.

[40] F. Chen, W. Ge, and J. Li. Molecular dynamics simulation of complex multiphase flow on a computer cluster with GPUs. *Sci. China, Ser. B: Chem.*, 52:372–380, 2009.

[41] E. Wu, Y. Liu, and X. Liu. An improved study of real-time fluid simulation on GPU. *Comp. Anim. Virtual Worlds*, 15, 2004.

[42] N. Goodnight, C. Woolley, G. Lewin, D. Luebke, and G. Humphreys. A multigrid solver for boundary value problems using programmable graphics hardware. In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 102–111. Eurographics Association, 2003.

[43] Z. Fan, F. Qiu, A. Kaufman, and S. Yoakum-Stover. GPU cluster for high performance computing. In *SC '04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, page 47. IEEE Computer Society, 2004.

[44] P. Blanchard, R. L. Devaney, and G. R. Hall. *Differential Equations*. Brooks/Cole Pub Co, 2006.

[45] R. Fernández, J. L. F. Abascal, and C. Vega. The melting point of ice I_h for common water models calculated from direct coexistence of the solid-liquid interface. *J. Chem. Phys.*, 124(14):144506, 2006.

[46] A. Brodsky. Is there predictive value in water computer simulations? *Chem. Phys. Lett.*, 261(4-5):563 – 568, 1996.

[47] D. van der Spoel, P. J. van Maaren, and H. J. C. Berendsen. A systematic study of water models for molecular simulation: Derivation of water models optimized for use with a reaction field. *J. Chem. Phys.*, 108(24):10220–10230, 1998.

[48] Y. Yonetani. Liquid water simulation: A critical examination of cutoff length. *J. Chem. Phys.*, 124(20):204501, 2006.

[49] D. Dubov and A. Vostrikov. Ordering of molecular dipole moments in a nanosized water cluster. *JETP Letters*, 88:674–678, 2008.

[50] B. A. Berg. Tip4p. (unpublished).

Acknowledgements

I want to thank Prof. Janke for the opportunity to do my master thesis in his work group. Prof. Janke as well as Prof. Berg supported me continuously with fruitful discussions and ideas, for which I am very thankful. Further I want to thank Stefan Horn for his help in programming questions and for introducing me to GPGPU computation. I am thankful for many inspiring conversations with Niklas Fricke, Sven Willner, Hannes Nagel, Reinke Isermann, Martin Treffkorn, and Heinz Seder.

In the end I want to thank my family and my fiancee for their continuous support.

Statement of authorship

I hereby certify that this master thesis has been composed by myself, and describes my own work, unless otherwise acknowledged in the text. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged. It has not been accepted in any previous application for a degree.

Johannes Zierenberg

After positive appraisal of this thesis, I agree that one copy of my presented thesis may remain at the disposal of the library of Leipzig University.

Johannes Zierenberg