

LUKAS WITTWER

**Molecular Dynamics Simulations of
Self Assembled Conjugated Thiols
on an Au(111) Surface**

DIPLOMARBEIT

zur Erlangung des akademischen Grades
Diplom-Ingenieur
Diplomstudium Technische Physik



Technische Universität Graz
Betreuer:

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Egbert Zojer

Institut für Festkörperphysik

Graz, Mai 2010

Acknowledgements

First of all I want to thank my supervisor *Prof. Egbert Zojer* for his support and expertise.

Furthermore, I would like to thank the people in our group, Anna, David, Ferdi, Gery, Oliver, Lukas, Marco, Thomas (vulgo Mulla), Simon and Stefan.

At this point, special thanks to Ferdi and Oliver for their expertise in compiling and other computer stuff and Gery for his tips and tricks in L^AT_EX.

Further, I want to thank to all my friends for their support throughout my studies.

Last but not least, I thank my parents and my whole family.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am

.....
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....
date

.....
(signature)

Abstract

The opportunity to modify surface properties by covering a substrate with self-assembled monolayers has led to increased interest in interfaces between metals and organic monolayers in the last years. While the electronic properties of such systems can be investigated in detail using density-functional theory (DFT) calculations, the investigation of structural and dynamic properties of is a prohibitive task within DFT. Molecular dynamics simulations (MD) allow for computer experiments at finite temperatures and reasonable system sizes, without intrinsically assuming crystallinity of the monolayer. Therefore, MD was employed to examine conformational properties of conjugated thiols on gold.

Furthermore, those simulations provide the possibility to investigate the dynamics of the thiols under the influence of vacancies on the SAM.

Kurzfassung

Die Möglichkeit die Eigenschaften von Metalloberflächen durch die Aufbringung von selbst assemblierten Monolagen zu verändern, hat zu verstärktem Interesse in der Beschreibung dieser Metall-Organik Grenzflächen geführt. Während im Rahmen der Dichtefunktionaltheorie (DFT) es möglich ist die elektronischen Eigenschaften solcher Systeme zu analysieren, können dynamische Eigenschaften und Strukturen damit nur schlecht bis gar nicht beschrieben werden. Molekulardynamikrechnungen erlauben es Untersuchungen bei endlichen Temperaturen und mit realistischen Systemgrößen durchzuführen, wodurch eine strenge kristalline Ordnung der Monolage nicht zwingend notwendig ist. Aus diesem Grund wurden Computersimulationen basierend auf Molekulardynamik durchgeführt um konformaive Eigenschaften von konjugierten Thiolen auf Gold zu untersuchen.

Weiters wurde der Einfluss von Leerstellen in der Monolage auf geometrische und dynamische Effekte hin untersucht.

Contents

Introduction	1
I. Fundamentals	3
1. Molecular dynamics	5
1.1. Time integration	5
1.1.1. The Verlet algorithm	6
2. The force field	9
2.1. Short range interactions	11
2.1.1. Bond stretching	11
2.1.2. Angle bending	12
2.1.3. Torsion	12
2.1.4. van der Waals interactions	12
2.1.5. Cross terms	13
2.1.6. Weighting factors	14
2.2. Long range interactions	15
2.2.1. The Ewald method	16
2.2.2. The particle-particle particle-mesh solver	18
3. Thermodynamical ensembles	23
3.1. Thermostats	23
3.1.1. Differential control	24
3.1.2. Proportional control	24
3.1.3. Stochastic controls	24
3.1.4. Integral controls	24
4. Atomic charges	27
II. Methodology	29
5. Overview	31
6. Charge assignment	35

7. Input data	39
7.1. The input file	39
7.2. Setting up the data file	43
8. Data evaluation	47
8.1. Angle distributions	47
8.1.1. Tilt angle	47
8.1.2. Twist angle	48
8.1.3. Azimuthal angle	48
8.2. Cumulative averages	49
8.3. Mean square displacement	49
8.4. Displacements	50
III. Results and discussion	53
9. Full coverage systems	55
9.1. Biphenyldithiol SAM on a Au(111) surface	55
9.1.1. Investigations at 300 K	55
9.1.2. Further investigations	61
9.2. Biphenylthiol SAM on a Au(111) surface	70
10. Low coverage systems	77
11. Negatively charged monolayers	85
Appendix	93
1. Input files	93
2. Data file	97
3. The build layer script	104
4. MATLAB codes	118
4.1. Angles	118
4.2. Displacements	127

List of Figures

2.1. Schematic representation of the contributions to the force field (FF) of Equation (2.1),[1]	10
2.2. Shape of the Morse potential (red line) and the harmonic potential (blue line)	11
2.3. Sketch of the Lennard-Jones potential	14
2.4. Graphical representation of the charge assignment function from Equation (2.24) up to the order $n = 5$	19
6.1. Saturated biphenylthiol molecule with AM1-BCC charges	36
6.2. Unsaturated biphenylthiol molecules with atomic charges determined by the AM1-BCC method	37
7.1. Auxiliary vetctors used in the coordinate transformation	42
7.2. Snapshot of a gold-biphenyldithiol system with 15×18 molecules arranged in a $(\sqrt{3} \times \sqrt{3}) - R30^\circ$ lattice	45
8.1. Schematic representation of the investigated angles,due the work of [2] . .	47
8.2. Graphical representation of the docking sited on a Au(111) surface. The light circles correspond to the top layer atoms, the dark to the lowest ones.	51
9.1. Saturated Biphenyldithiol (a) and with cut off hydrogen at the docking site (b). The partial charges were derived using the AM1-BCC assignment scheme	57
9.2. Distribution of the tilt angle Θ recorded during equilibration in the NVE ensemble for 100 ps at 300 K	58
9.3. Distribution of the twist angle Ψ recorded during equilibration in the NVE ensemble for 100 ps at 300 K	58
9.4. Distribution of the azimuthal angle ϕ recorded during equilibration in the NVE ensemble for 100 ps at 300 K	59
9.5. Distribution of occupied docking sites of certain timesteps during NVE equilibration at 300 K.	59
9.6. Histogram of the displacement of the docking atoms during NVE equilibration at 300 K	60
9.7. Averaged angles versus time during NVE equilibration at 300 K	60
9.8. Mean square displacement of the docking atoms during NVE equilibration at 300 K	61

List of Figures

9.9. Developing of the temperature (upper graph) and the total energy E_{toto} (lower graph) during a simulation.	62
9.10. Cumulative averages of the tilt (Θ), twist (Ψ) and azimuthal (ϕ) angle at 300 K and 200 K	63
9.11. Cumulative averages of the tilt (Θ), twist (Ψ) and azimuthal (ϕ) angle at 150 K and 100 K	64
9.12. Distributions of the (a) tilt, (b) twist and (c) azimuthal angle of biphenyldithiol for different temperatures	65
9.13. Standard deviation of the distribution of the (a) tilt, (b) twist and (c) azimuthal angle of biphenyldithiol plotted versus temperature	67
9.14. Comparison of the mean square displacement (MSD) of the docking atoms at different temperatures of a biphenyldithiol self assembled monolayer (SAM)	68
9.15. Snapshots of the simulation cell during NVE equilibration of biphenyldithiol on gold at different temperatures; only the carbon rings and sulfur atoms of the SAM molecules are drawn.	69
9.16. Distributions of the (a) tilt, (b) twist and (c) azimuthal angle of biphenylthiol for different temperatures	71
9.17. Standard deviation of the distribution of the (a) tilt, (b) twist and (c) azimuthal angle of biphenylthiol plotted versus temperature	72
9.18. Comparison of the MSD of the docking atoms of a biphenylthiol SAM at different temperatures	73
9.19. Occupied docking sites of the biphenylthiol monolayer during NVE equilibration at 300 K	73
9.20. Histogram of the displacements of the docking atoms during NVE equilibration at 300 K for biphenylthiol	74
9.21. Snapshots of the simulation cell of biphenylthiol on gold during NVE equilibration at different temperatures; only the carbon rings and sulfur atoms of the SAM molecules are drawn.	75
10.1. Distributions of the tilt angle of a biphenyldithiol monolayer (a) and a biphenylthiol monolayer (b) for 90% coverage of the surface. The dashed lines represent the distributions of fully covered systems. In picture (a) the peak representing the lying molecules is encircled.	78
10.2. Distributions of the twist angles of a biphenyldithiol monolayer (a) and a biphenylthiol monolayer (b) for 90% coverage of the surface. The dashed lines represent the distributions of fully covered systems.	78
10.3. Distributions of the azimuthal angles of a biphenyldithiol monolayer (a) and a biphenylthiol monolayer (b) for 90% coverage of the surface. The dashed lines represent the distributions of fully covered systems.	79
10.4. Mean square displacements of the thiol docking groups in a 90 % covered monolayer of (a) biphenyldithiol and (b) biphenylthiol. The dashed lines indicate the MSD of the full coverage systems.	81

10.5. Occupied docking sites of a (a) biphenylthiol and a (b) biphenyldithiol monolayer of 90 % coverage during NVE equilibration at 300 K.	82
10.6. Distribution of the docking atoms' displacement during NVE equilibration at 300 K of a 90 % coverage SAM of (a) biphenylthiol and (b) biphenyldithiol. The vertical dotted lines indicate the distances between the described docking sites	83
10.7. Snapshots of the simulation cell of (a) biphenylthiol and (b) biphenyldithiol SAMs with 90 % coverage. The encircled regions in snapshot (a) refer to those molecules resulting in the third peak in the twist angle distribution (Figure 10.2). The lying biphenyldithiol molecules are marked by circles (solid line) in picture (b). The dashed circles in picture (b) mark the domains with preserved herringbone structure.	84
11.1. Angle distributions of a neutral and negative biphenyldithiol SAM at full coverage.	86
11.2. Angle distributions of a neutral and negative biphenyldithiol SAM at 90 % coverage.	87

Introduction

Surface modified metals have gained growing interests over recent in areas years such as electronics or for making metals resistant to corrosion. In terms of electronics, growing self assembled monolayers (SAM) on metals allow tuning their work function and therefore modifying the properties of an electronic device, e.g. by inserting a monolayer between the gold electrodes and the organic semiconductor in a transistor [3]. A way to handle those modifications in work function computationally is by use of density functional theory (DFT). Although DFT simulations have been applied successfully on systems of thiols on gold surfaces in order to obtain quantities such as the shift in the work function, the density of states, or orbital energies, they have drawbacks when it comes to describing the dynamics of a system at finite temperatures. Since the method of DFT is only restricted to temperatures of 0 K different techniques must be employed to investigate dynamical effects of SAMs, for example the influence of temperature on the disorder of the system.

Molecular dynamics allow for investigations at finite temperatures and therefore to examine the temperature dependence of quantities such as disorder within a monolayer. Thus a major objective of this work was to investigate SAMs of conjugated thiolates on a Au(111) surface by means of molecular dynamics (MD). In order to do so, appropriate MD software had to be chosen and tested by reproducing data from literature. A further step was then to investigate full coverage SAMs at different temperatures and finally to examine the impacts of voids in the monolayer. Since in this work classical MD is used for the first time in our group in order to investigate SAMs, it is focused on theoretical and methodological aspects rather than on actual results.

In the first part of this work a brief introduction is given in which the most fundamental principles of MD are discussed and the theoretical background of the used algorithms is shortly provided.

The second part is focused on the methodology, i.e., the used software and algorithms, the simulation setup, and the simulation steps are described in detail. The main features

List of Figures

of an input file for the MD software are listed, too.

In the final section the obtained results are presented and discussed. First, the reproduced data at 300 K are compared to those found in literature [2], after that the results of calculations at different temperatures (200 K, 150 K, 100 K) are provided. Such calculations were done for two systems, namely biphenylthiolate and biphenyldithiolate on gold and the results are then compared. Furthermore, the outcome of calculations of these two systems at lower coverages is presented. For those low coverage systems the spotlight is on the analysis related to the displacements of the docking groups.

Part I.

Fundamentals

1. Molecular dynamics

Literature: Refs. [4] [1]

The basic idea of classical molecular dynamics (MD) is to solve the classical equations of motions

$$\mathbf{f}_i = m_i \ddot{\mathbf{r}}_i \quad (1.1a)$$

$$\mathbf{f}_i = -\nabla U(\mathbf{r}_i). \quad (1.1b)$$

step by step numerically for each particle (see below). Here \mathbf{f}^i denotes the force acting on the i th particle at position $\mathbf{r}_i = (r_{1,i}, r_{2,i}, r_{3,i})$ and $U(\mathbf{r}_i)$ the potential energy.

With $\mathbf{r}^N = (\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^N)$ being the position of all particles in the system, $3N$ equations of motions have to be solved in total at each step.

The shape of the potential depends on how the particles interact with each other, e.g. bonded or non-bonded interactions, electrostatic terms etc.

The equations of motion are then solved iteratively by use of certain time integration algorithms (e.g. Verlet algorithm).

1.1. Time integration

Literature: Refs. [1]

In MD the equations of motion are solved step-by-step, i.e. time integration is not done continuously but by some integration scheme with finite steps of δt . In the limit of δt going to zero the applied scheme should theoretically produce the exact trajectory of the system. Furthermore, integrating the equations of motion by means of a finite integrator must not violate energy conservation and small perturbations should not result in instabilities of the system.

1.1.1. The Verlet algorithm

The simplest approach for a finite integration scheme is to expand the positions and velocities in a Taylor series for a sufficiently small δt :

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + \dot{\mathbf{r}}(t)\delta t + \frac{1}{2}\ddot{\mathbf{r}}\delta t^2 + \frac{1}{6}\mathbf{c}\delta t^3 \dots \quad (1.2)$$

$$\mathbf{v}(t + \delta t) = \dot{\mathbf{r}}(t + \delta t) = \mathbf{v}(t) + \dot{\mathbf{v}}(t)\delta t + \frac{1}{2}\ddot{\mathbf{v}}\delta t^2 + \frac{1}{6}\mathbf{d}\delta t^3 \dots \quad (1.3)$$

Expanding with respect to $-\delta t$ yields on the other hand:

$$\mathbf{r}(t - \delta t) = \mathbf{r}(t) - \dot{\mathbf{r}}(t)\delta t + \frac{1}{2}\ddot{\mathbf{r}}\delta t^2 - \frac{1}{6}\mathbf{c}\delta t^3 \pm \dots \quad (1.4)$$

$$\mathbf{v}(t - \delta t) = \dot{\mathbf{r}}(t - \delta t) = \mathbf{v}(t) - \dot{\mathbf{v}}(t)\delta t + \frac{1}{2}\ddot{\mathbf{v}}\delta t^2 - \frac{1}{6}\mathbf{d}\delta t^3 \pm \dots \quad (1.5)$$

with \mathbf{c} and \mathbf{d} denoting the third derivatives of the position and the velocity respectively. Combining Equations (1.2) and (1.4) and Equations (1.3) and (1.5) gives

$$\mathbf{r}(t + \delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \delta t) + \ddot{\mathbf{r}}\delta t^2 + \mathcal{O}(t^4) \quad (1.6)$$

$$\mathbf{v}(t + \delta t) = 2\mathbf{v}(t) - \mathbf{v}(t - \delta t) + \ddot{\mathbf{v}}\delta t^2 + \mathcal{O}(t^4) \quad (1.7)$$

The cruxes of Equation (1.7) is on the one hand, that one has to evaluate the third derivative of the position in Equation (1.7). Furthermore, in order to compute the positions and velocities for the next time step, information on the previous two steps must be stored. Since the third derivative of the position is normally not done in a MD run one has to modify Equation (1.7) somewhat to avoid third derivatives. This can be done by subtracting Equation (1.2) from (1.4) leading to an expression for the velocity without third derivatives

$$\mathbf{v}(t) = \frac{1}{2\delta t}(\mathbf{r}(t + \delta t) - \mathbf{r}(t - \delta t)) + \mathcal{O}(t^3) \quad (1.8)$$

which is however not as accurate as Equation (1.7) and it does not provide information about velocities at $t + \delta t$.

The method presented above is the so called *Verlet algorithm* [5] which has still the drawback that information of the previous two steps has to be stored to compute the next one.

An optimization in terms of storage can be achieved by reformulating Equations (1.2)

and (1.4) leading to a modified Verlet integrator the so called *velocity Verlet* algorithm:

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + \dot{\mathbf{r}}(t)\delta t + \frac{1}{2}\ddot{\mathbf{r}}\delta t^2 \quad (1.9)$$

$$\mathbf{v}(t + \delta t) = \mathbf{v}(t) + \frac{1}{2}\delta t(\dot{\mathbf{v}}(t) + \dot{\mathbf{v}}(t + \delta t)) \quad (1.10)$$

Equation (1.10) contains the acceleration at the time $t + \delta t$ which cannot be calculated directly but via the following intermediate step. First the velocities are calculated at $t + \delta t/2$ by

$$\mathbf{v}(t + \delta t/2) = \mathbf{v}(t) + \frac{1}{2}\delta t\dot{\mathbf{v}}(t), \quad (1.11)$$

next the forces acting on the particles at $t + \delta t$ are evaluated yielding $\dot{\mathbf{v}}(t + \delta t)$ which is eventually used to compute

$$\mathbf{v}(t + \delta t) = \mathbf{v}(t + \delta t/2) + \frac{1}{2}\delta t\dot{\mathbf{v}}(t + \delta t). \quad (1.12)$$

One advantage of the velocity Verlet algorithm compared to the normal Verlet method is that the positions and velocities are updated simultaneously thus allowing to calculate the kinetic energy without a time delay of δt .

2. The force field

Literature: Refs. [4] [1]

In molecular dynamics the concept of force fields is used to compute mostly complex interactions between atoms by approximating the potential energy with a potential function being only dependent on the relative position of the considered atoms. Instead of explicitly solving the Schrödinger equation, which becomes impossible for large unit cells, the interaction energy is split in single components related to certain kinds of interaction. The term FF is related to the shape of the equations on the one hand as well as to the numeric values of the parameters in these equations on the other hand. Since all relevant quantum-mechanical effects are taken into account when evaluating the parameters for the FF, the particles or atoms are assumed to be point-shaped in a MD simulation. Usually there are terms for bond stretching, for angle bending, bond rotation (torsion) and for non-bonded interactions (van der Waals (vdW) and electrostatic), contributing to the total potential energy of the system. An example, in this case the AMBER FF [6], of the functional form of a FF is shown in Equation (2.1).

$$U(\mathbf{r}^N) = \sum_{\text{bonds}} \frac{1}{2} k_b (l - l_0)^2 + \sum_{\text{angles}} k_a (\theta - \theta_0)^2 + \sum_{\text{torsions}} \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)] + \sum_{j=1}^{N-1} \sum_{i=j+1}^N \left\{ \epsilon_{i,j} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\} \quad (2.1)$$

Here l describes the deviation from the equilibrium bond length l_0 , θ is the according deviation from the equilibrium angle θ_0 . The third term in Equation (2.1) describes the rotational energy change of the bond. The rotation of bondings is usually referred as dihedrals. By the last two terms the non-bonding interactions are modeled by a Lennard-Jones potential with a Coulomb term for electrostatic interactions. The individual contributions to the FF are depicted in Figure (2.1). The parameters in Equation (2.1) are specific for a certain type of atom and how this atom is in relation to others.

2. The force field


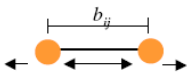
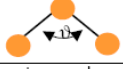

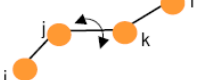
6-9 van der Waals	
6-12 van der Waals	
Electrostatic	
Quadratic bond-stretching	
Morse bond-stretching	
Bond-bending	
Improper dihedrals	
Proper dihedrals	

Figure 2.1.: Schematic representation of the contributions to the FF of Equation (2.1),[1]

In the case of a bonded atom, several effects have to be taken into account, e.g., the hybridization or to what other atoms it is bonded to. The bond-stretching parameters for a carbon - carbon bond for example depend on whether the bond is in an aromatic ring or in an aliphatic compound. Generally the parameters for bonded interactions are explicitly given for the involved atoms.

Parametrization itself can be a very tedious task; the values for the certain parameters can be, for example, obtained from Monte Carlo simulations, density functional theory calculations or by fitting to experiments.

Another key feature of a proper FF is transferability. This means that a set of parameters for a specific type of interaction should be applicable to a variety of systems. It should be mentioned that most FFs are parameterized for certain classes of molecules. This has to be taken into account when choosing a certain FF for a particular problem.

Basically one can distinguish among two kinds of interactions described by certain force fields, namely:

- short range and
- long range

interactions.

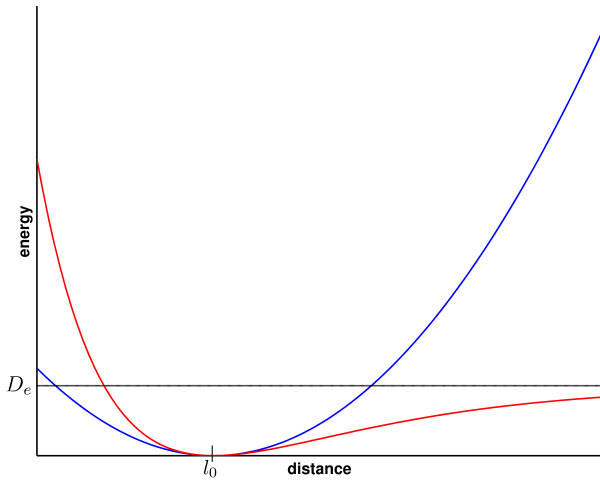


Figure 2.2.: Shape of the Morse potential (red line) and the harmonic potential (blue line)

2.1. Short range interactions

2.1.1. Bond stretching

The simplest method to model a bond is the harmonic potential

$$u_b = \frac{k_b}{2}(l - l_0)^2 \quad (2.2)$$

with l the deviation from the equilibrium bond length l_0 and k_b the stretching constant.

A more sophisticated approximation is the so called Morse potential

$$u_b = D_e[1 - e^{-a(l-l_0)}]^2. \quad (2.3)$$

Around the equilibrium distance Morse and the harmonic potential are equivalent but for larger deviations from the equilibrium the Morse potential allows for dissociation and therefore bond breaking. However parameterizing the Morse potential is a more difficult task due to the three parameters to be determined (D_e, a, l_0).

In Figure 2.2 the shape of the harmonic and the Morse potential are sketched.

2. The force field

2.1.2. Angle bending

A common way to describe the energy of angle bending is again a harmonic potential

$$u_a = \frac{k_a}{2}(\theta - \theta_0)^2, \quad (2.4)$$

with θ being the deviation from some reference angle θ_0 and k_a a constant.

2.1.3. Torsion

Usually the rotation of a bond is expressed by a cosine series expansion:

$$u_t = \sum_{n=0}^N \frac{V_n}{2} [1 + \cos(n\omega - \gamma)]. \quad (2.5)$$

Here V_n is related to the 'barrier height', n the multiplicity, i.e. it gives the number of minima, and γ a phase factor. In conjunction with n it determines, where the potential has its minima.

2.1.4. van der Waals interactions

Besides, the discussed bonded interactions there are also interactions not being related to bonds, namely the vdW interaction and electrostatic interactions. In contrast to electrostatic, vdW interactions also occur among uncharged particles and show attractive behavior.

The attractive characteristics can be understood by means of fluctuation in the electron clouds of the atoms forming instantaneous dipoles. Those dipoles may induce dipoles in neighboring atoms resulting in an attractive force due to dipole-dipole interactions.

In order to find an appropriate potential curve for describing vdW forces, the potential must be repulsive for short distances. This repulsive behavior is usually not denoted as vdW interaction. It lies in the Pauli principle, stating that two electrons must not be in the same quantum state. Thus, two electrons with parallel spin occupying the same region in space repel each other, therefore those repulsive forces are sometimes referred to as *exchange forces*.

The most common approximation to model vdW interactions is the Lennard-Jones

potential [7], describing the interaction potential between two particles as follows:

$$u_{ij} = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]. \quad (2.6)$$

The variable σ_{ij} in Equation (2.6) denotes the so called collision diameters indicating the distance at which the energy is zero, ϵ_{ij} is the depth of the potential well and r_{ij} is the distance between the considered atoms. Equation (2.6) can be rewritten in terms of the distance r^m at which the potential has its minimum:

$$u_{ij} = \epsilon_{ij} \left[\left(\frac{r_{ij}^m}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{ij}^m}{r_{ij}} \right)^6 \right]. \quad (2.7)$$

The shape of the Lennard-Jones potential is shown in Figure 2.3. For large distances it quickly converges to zero. Thus neglecting atoms beyond a cutoff radius r_c is legitimate.

In the case of the Lennard-Jones potential, the FF parameters for different types of atoms may be determined, if not explicitly given, by applying so called *mixing rules*. A widely used set of such rules are the Lorentz-Berthelot mixing rules [8], using the arithmetic mean to evaluate σ_{AB}

$$\sigma_{AB} = \frac{\sigma_A + \sigma_B}{2} \quad (2.8)$$

with σ_A and σ_B the parameters for atom type A and B respectively. The depth of the potential well ϵ_{AB} is obtained by forming the geometric mean of the pure atomic parameters ϵ_A and ϵ_B :

$$\epsilon_{AB} = \sqrt{\epsilon_A \epsilon_B}. \quad (2.9)$$

Another commonly used shape of the vdW potential is the Buckingham potential [9], in which the repulsive part is described by an exponential term:

$$u_{ij} = A_{ij} e^{\frac{r}{\rho_{ij}}} - C_{ij} r^{-6} \quad (2.10)$$

2.1.5. Cross terms

In some FF cross terms take into account that the interactions described above influence each other, e.g. if in a three atom system a bond angle is decreased the associated bonds stretch in order to reduce the interaction of the atoms separated by two bonds. Atoms

2. The force field

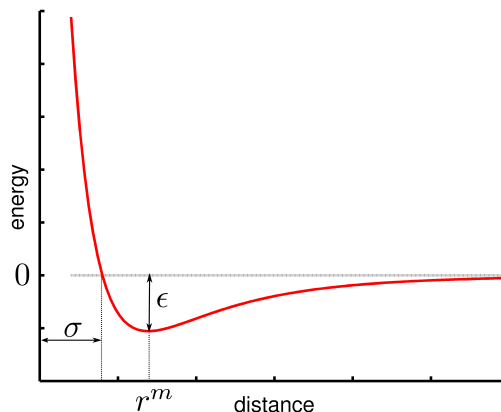


Figure 2.3.: *Sketch of the Lennard-Jones potential*

that are separated by two bonds are commonly denoted as 1,3 atom pairs, those separated by one bond 1,2 atom pairs and those by three bonds 1,4 pairs. Actually one has to take into account cross terms between all interactions in a system but for many applications (e.g. predicting structural properties), it is sufficient to define cross terms for two internal coordinates, i.e. stretch-stretch, stretch-torsion etc.

The AMBER [6] and DREIDING [8] force field used in this work are so called 'Class I' force fields, i.e., there are not any cross terms. The brief description above is just given for the sake of completeness.

2.1.6. Weighting factors

The bonded terms in a FF do not fully represent the interaction between the considered particles. There is always a vdW like contribution or in case of charged particles there are additional electrostatic interactions. These non-bonded effects may be taken into account when determining the FF parameters, or by weighting the potential energy of such non-bonded interactions by a set of factors. These factors indicate to what extent electrostatic or vdW interactions are considered when determining the potential energy between two particles. Normally there is a set of three parameters for the 1,2, 1,3 and 1,4 pairs respectively, e.g. a set of (0,0,0.5) for vdW interactions means that vdW interactions are turned off for 1,2 and 1,3 pairs, for 1,4 pairs the vdW energy is multiplied by a factor 0.5.

2.2. Long range interactions

Literature: Refs. [10], [11] [12]

In atomic systems, electrostatic charges interact via a Coulomb potential being inverse-proportional to the distance ($u_{\text{coul}} \propto \frac{1}{r}$). Thus, the potential energy decays quite slowly for large distances and, therefore, one must take into account all particles in the system because truncating beyond a certain distance would lead to a significant error. Furthermore, the Coulomb potential strongly varies for short distances leading to additional difficulties when tackling it computationally. Those strong variations prohibit a representation by the first few term of its Fourier series.

For systems with open boundary conditions the Coulomb force acting between two charges can be evaluated straight forwardly by summing over all atomic pairs

$$u_{\text{coul}} = \sum_{i,j=1}^N \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \quad (2.11a)$$

$$F_{\text{coul}} = \sum_{i,j=1}^N \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}^2} \quad (2.11b)$$

with N the total number total number of particles in the system, leading to a problem of the order $\mathcal{O}(N^2)$.

For periodic systems the situation is different because in this case beside the interactions in the *central cell* also the interactions between the central and all the periodic images must be considered. Thus Equation (2.11a) becomes

$$u_{\text{coul}} = \frac{1}{2} \sum_{\mathbf{n} \in \mathbb{Z}^3} \sum_{i,j=1}^N \frac{q_i q_j}{4\pi\epsilon_0 |\mathbf{r}_{ij} + \mathbf{n}L|} \quad (2.12)$$

here L is the length of the simulation box. Unfortunately the sum in Equation (2.12) converges quite slowly making a direct evaluation almost impossible. Thus more sophisticated algorithms must be applied.

2.2.1. The Ewald method

The basic idea of the Ewald method [13] is to split the sum in Equation (2.12) into two parts, namely

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1-f(r)}{r} \quad (2.13)$$

whereas $f(r)$ should meet the following requirements when applying it on Equation (2.12):

- For large distances beyond a cutoff radius r_{max} the term $\frac{f(r)}{r}$ should be sufficiently small so that it can be neglected. Thus, summing up to the cutoff yields a good approximation for the potential at small distances.
- The $\frac{1-f(r)}{r}$ has to be smooth enough at all distances to represent its Fourier transform only by a few k-vectors $\mathbf{k} < k_{max}$, therefore, allowing to calculate this contribution to the Coulomb potential in k-space.

Traditionally, the complimentary error function

$$\text{erfc}(r) = \frac{2}{\sqrt{\pi}} \int_r^\infty dt e^{-t^2} \quad (2.14)$$

is chosen as f . Inserting Equation (2.14) in Equation (2.13) and Equation (2.12) respectively leads to the following expression for the electrostatic potential

$$U = U^{(r)} + U^{(k)} + U^{(s)} + U^{(d)} \quad (2.15)$$

with $U^{(r)}$ the contribution from real space summation, $U^{(k)}$ the contribution from the summation in reciprocal space, $U^{(s)}$ the self energy and $U^{(d)}$ the dipole correction. In detail the terms in Equation (2.15) look as follows:

$$U^{(r)} = \frac{1}{2} \sum_{\mathbf{m} \in \mathbb{Z}} \sum_{i,j} q_i q_j \frac{\text{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{m}L|)}{|\mathbf{r}_{ij} + \mathbf{m}L|} \quad (2.16a)$$

$$U^{(k)} = \frac{1}{2} \frac{1}{V} \sum_{\mathbf{k} \neq \mathbf{0}} \frac{4\pi}{k^2} e^{-k^2/4\alpha^2} |\tilde{\rho}(\mathbf{k})| \quad (2.16b)$$

$$U^{(s)} = -\frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2 \quad (2.16c)$$

$$U^{(d)} = \frac{2\pi}{(1+2\epsilon')V} \left(\sum_i q_i \mathbf{r}_i \right)^2 \quad (2.16d)$$

Here $\tilde{\rho}(\mathbf{k})$ denotes the Fourier transformed charge density which is defined as

$$\tilde{\rho}(\mathbf{k}) = \int_V d^3r \rho(\mathbf{k}) e^{-i\mathbf{k}\mathbf{r}} = \sum_{j=1}^N q_j e^{-i\mathbf{k}\mathbf{r}_j}. \quad (2.17)$$

Equation (2.16d) arises from an exact mathematical treatment of the conditional convergences of Equation (2.12), for a detailed description see [11]. The parameter α in the equations above can be understood as a tuning factor between real and reciprocal space sometimes called *Ewald parameter*. The sums in Equations (2.16a) and (2.16b) converge exponentially thus permitting small cutoffs for r and k , respectively. Strictly speaking the Ewald method cannot be applied to systems bearing a net-charge unequal to zero, because in this case the sum in Equation (2.12) diverges. To overcome that one can introduce a homogeneous charge background making the system charge neutral. This requires an additional term in Equation (2.15) of the form

$$U^{(n)} = -\frac{\pi}{2\alpha^2 V} \left(\sum_i q_i \right)^2. \quad (2.18)$$

In order to obtain the force acting on a particle one has to calculate the spatial derivatives of the potential:

$$\mathbf{F}_i = -\frac{\partial}{\partial \mathbf{r}_i} E \quad (2.19)$$

leading to the following expressions:

$$\begin{aligned} \mathbf{F}_i^{(r)} = & q_i \sum_j q_j \sum_{\mathbf{m} \in \mathbb{Z}} \left(\frac{2\alpha}{\sqrt{\pi}} \exp(-\alpha^2 |\mathbf{r}_{ij} + \mathbf{m}L|^2) \right. \\ & \left. + \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{m}L|)}{|\mathbf{r}_{ij} + \mathbf{m}L|} \right) \frac{\mathbf{r}_{ij} + \mathbf{m}L}{|\mathbf{r}_{ij} + \mathbf{m}L|^2} \end{aligned} \quad (2.20)$$

$$\mathbf{F}_i^{(k)} = q_i \sum_j q_j \frac{1}{V} \sum_{\mathbf{k} \neq \mathbf{0}} \frac{4\pi \mathbf{k}}{k^2} \exp\left(-\frac{k^2}{4\alpha^2}\right) \sin(\mathbf{k}\mathbf{r}_{ij}) \quad (2.21)$$

$$\mathbf{F}_i^{(k)} = -\frac{4\pi q_i}{(1 + 2\epsilon')V} \sum_i q_j \mathbf{r}_j \quad (2.22)$$

Unfortunately, the the Fourier transforms in the context of Equations (2.16b) and (2.21) are extremely time consuming procedures. For optimized values of k_{max} and r_{max} the computational effort can be reduced to the order $\mathcal{O}(N^{3/2})$, which for large systems is still expensive.

2. The force field

Further improvement can be achieved by applying the fast Fourier transform (FFT) algorithm instead of the ordinary Fourier transforms reducing the effort for the operations in k-space to the order $\mathcal{O}(N \log N)$. For sufficiently large systems the cutoff in real space can be chosen in such a fashion that computational efforts for the entire Ewald sum scale with $N \log N$. The crucial point in the context of FFT lies in the fact that it is a discrete method and, therefore, the reciprocal part of the Ewald sum has to be discretized bearing several problems. The discretization is done by dividing the simulation cell in a fixed mesh with a certain number of grid points in each direction in space. Given a particular mesh the following steps are carried out:

1. First of all the point charges with continuous position vectors \mathbf{r}_i must be assigned to charge densities at the fixed grid points.
2. The next step is to compute the electric field on the mesh points applying FFT.
3. Eventually, the forces on each particle are calculated.

2.2.2. The particle-particle particle-mesh solver

The steps listed above haven been implemented in various styles, the most famous one is the particle-particle particle-mesh (P3M) method developed by Hockney and Eastwood [14]. The point charges are assigned to the grid points by using a grid based charge density

$$\rho_m(\mathbf{r}_p) = \frac{1}{h^3} \sum_{i=1}^N q_i W(\mathbf{r}_p - \mathbf{r}_i) \quad (2.23)$$

where h is the mesh spacing and \mathbf{r}_p the position of a grid point. The function W in Equation (2.23) is the so called charge assignment function. In the framework of the P3M solver, cardinal B-splines are chosen for W which can be obtained via their general definition

$$W_n(r) = \sum_{l=0}^{n-1} A_n(l, j) \left(x - \frac{j}{2}\right)^l \text{ for } -\frac{1}{2} < r - \frac{j}{2} < \frac{1}{2} \quad (2.24)$$

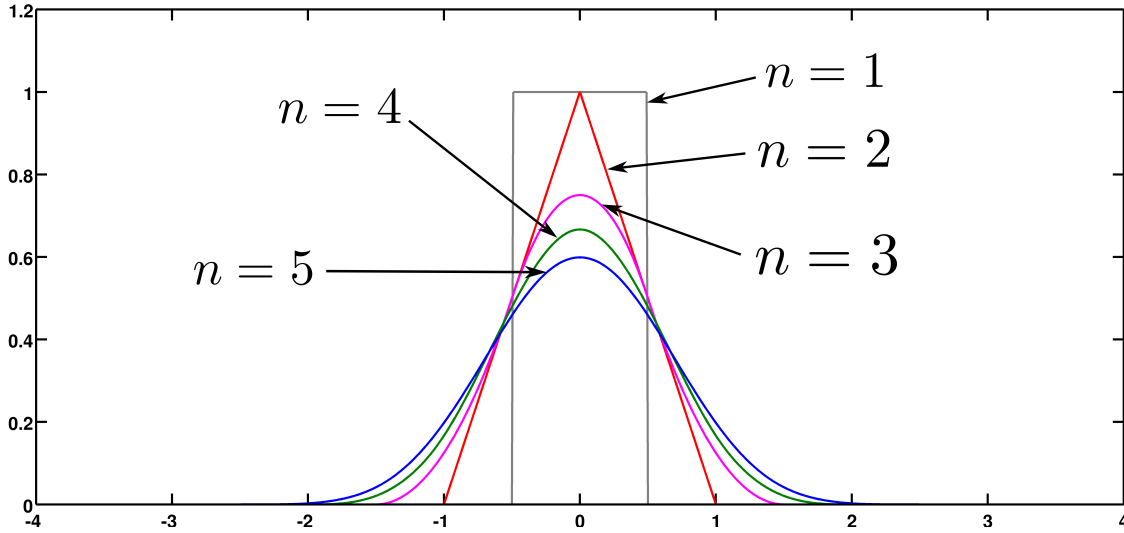


Figure 2.4.: Graphical representation of the charge assignment function from Equation (2.24) up to the order $n = 5$

with

$$\begin{aligned}
 A_{n+1}(l+1, j) &= \frac{A_n(l, j+1) - A_n(l, j-1)}{l+1} \\
 A_{n+1}(0, j) &= \sum_{l=0}^{n-1} 2^{-l} \frac{A_n(l, j-1) + (-1)^l A_n(l, j+1)}{l+1} \\
 A_1(0, 0) &= 1
 \end{aligned} \tag{2.25}$$

with j running from $-(n-1)$ to $(n-1)$ in steps of 2 and n denoting the order of the spline. In Figure 2.4 the charge assignment function is depicted up to the order 5, the position of the peak corresponds to the position of a grid point. The order in Equation (2.24) determines over how many grid points in each direction the charges are distributed thus the peaks in Figure 2.4 get broader at higher n .

The electrostatic potential on the other hand can be linked to the charge density by the Poisson equation, which has the integral form

$$\Phi(\mathbf{r}) = \int \int \int G(\mathbf{r} - \mathbf{r}') \rho(\mathbf{r}') d\mathbf{r}'. \tag{2.26}$$

In terms of mathematics, Equation (2.26) is a convolution which can formally be written

2. The force field

as $G * \rho$ and can be evaluated by means of Fourier transform yielding

$$\tilde{\Phi} = \tilde{G}\tilde{\rho}. \quad (2.27)$$

To calculate the electric field one has to calculate the gradient of the electrostatic potential

$$\mathbf{E} = -\nabla\Phi$$

leading in k-space to a multiplication with the factor $i\mathbf{k}$. Thus, the electric field can be rewritten as

$$\begin{aligned} \tilde{E}(\mathbf{k}) &= i\mathbf{k}\tilde{\rho}_m\hat{G}_{opt} \\ &\Downarrow \\ E(\mathbf{r}_p) &= \overleftarrow{\text{FFT}}(\tilde{E}(\mathbf{k})). \end{aligned} \quad (2.28)$$

The Fourier transform of Equation (2.23) yields the charge density in reciprocal space which reads as follows

$$\tilde{W}(\mathbf{k}) = h^3 \left(\frac{\sin(\frac{1}{2}k_x h)}{\frac{1}{2}k_x h} - \frac{\sin(\frac{1}{2}k_y h)}{\frac{1}{2}k_y h} - \frac{\sin(\frac{1}{2}k_z h)}{\frac{1}{2}k_z h} \right)^n. \quad (2.29)$$

In Equation (2.28) \hat{G}_{opt} denotes an optimized Greens function reducing the error of the grid approximation to a minimum, it is written as follows

$$\hat{G}_{opt} = \frac{i\mathbf{k} \sum_{\mathbf{m} \in \mathbb{Z}} \tilde{U}^2(\mathbf{k} + \frac{2\pi}{h}) \tilde{\mathbf{R}}(\mathbf{k} + \frac{2\pi}{h})}{|\mathbf{k}|^2 \left[\sum_{\mathbf{m} \in \mathbb{Z}} \tilde{U}^2(\mathbf{k} + \frac{2\pi}{h}) \right]^2} \quad (2.30)$$

$$\text{with } \tilde{\mathbf{R}}(\mathbf{k}) := -i\mathbf{k} \frac{4\pi}{k^2} e^{-k^2/4\alpha^2}$$

$$\tilde{U}(\mathbf{k}) := \frac{\tilde{W}(\mathbf{k})}{h^3}.$$

Having calculated the electric field, eventually one can determine the electrostatic force acting on each particle:

$$\mathbf{F}_i = q_i \sum_{\mathbf{r}_p} \mathbf{E}(\mathbf{r}_p) W(\mathbf{r}_i - \mathbf{r}_p) \quad (2.31)$$

Finally, it should be mentioned that there is a way to determine an optimal number of mesh points at a given precision or, more precisely, at a given tolerance of the error in force made during the mesh calculation. For a detailed derivation see [15]. In principle this method evaluates an optimized splitting parameter α depending on the desired precision. With this optimal α , \hat{G}_{opt} is determined and finally one ends up with the optimized set of grid points.

3. Thermodynamical ensembles

Literature: Refs. [1], [16]

When the equations of motion as described in Chapter 1 are solved without any further restrictions one ends up sampling a microcanonical ensemble. The partition function of a microcanonical ensemble is defined by the number of particles N , the volume V and the kinetic energy E . A system referred to as a microcanonical ensemble is in thermodynamic equilibrium at a given Energy and it has to be closed, i.e. the number of particles and volume is constant.

Furthermore, MD allows for investigations at a given temperature or pressure, by applying thermostats or barostats, respectively. Since in the context of this work just examinations at given temperatures were conducted, the following description is restricted to thermal control algorithms.

Instead of controlling the kinetic energy of a system, one can control its temperature in order to move it to the thermodynamic equilibrium. In terms of physics, such a system may be illustrated by a heat bath of a given temperature, thus the considered system is defined by the number of particles N , its volume V and the temperature T .

3.1. Thermostats

As mentioned before, solving the equations of motion in its original form leads to a microcanonical ensemble. To control temperature during a MD run, there are several mechanisms that may be applied in addition to solving the equations of motion. Most of these methods make use of the fact that the temperature is proportional to the particle velocities and momenta respectively via

$$T = \frac{m\overline{v^2}}{3R} \quad (3.1)$$

with R the gas constant and m the mass of the considered particle.

3. Thermodynamical ensembles

3.1.1. Differential control

The simplest way to control temperature is by rescaling the momenta of particles in order to obtain the desired temperature T_0

$$\mathbf{p}_i \Rightarrow \sqrt{\frac{T_0}{T}} \mathbf{p}_i \quad (3.2)$$

with T , the actual temperature, calculated from the velocities using Equation (3.1)

3.1.2. Proportional control

Another, more proper method to temper a system is by rescaling the particle velocities in a more sophisticated way. In contrast to the differential thermostat the proportional thermostat does not rescale the velocities exactly to the desired temperature but it 'moves' the system towards the target temperature thus permitting some slight fluctuations around T_0 . To achieve that, the velocities are multiplied by a rescaling factor λ given by

$$\lambda = \left[1 + \frac{\delta t}{\tau_T} \left(\frac{T_0}{T} - 1 \right) \right]^{\frac{1}{2}} \quad (3.3)$$

with τ_T the coupling constant indicating how long it takes the system to reach T_0 . This method was first introduced by Berendsen et al. [17]. It describes the weak coupling to an external heat bath.

3.1.3. Stochastic controls

The idea behind a stochastic thermostat is to introduce virtual particles colliding with the system particles and, thereby, controlling the temperature. To reach the target temperature, T_0 , the virtual particles obey a Maxwell distribution according to T_0 .

3.1.4. Integral controls

In the case of integral thermostats, the coupling to an external heat bath of the a certain temperature is realized by introducing an additional degree of freedom. The coupling between the heat reservoir and the system is modeled by scaling the particles' velocities. In order to do so the variables of the system are transformed to a virtual system by

$$d\tau/dt = s,$$

where τ is the virtual time and s a scaling factor. The transformed momenta and spatial coordinates read as follows.

$$\mathbf{p}_i = \boldsymbol{\pi}_i s, \quad \mathbf{q}_i = \boldsymbol{\rho}_i. \quad (3.4)$$

leading to a Hamiltonian of the virtual system

$$H^v = \sum_{i=1}^N \frac{\boldsymbol{\pi}_i^2}{2m_i s^2} + U(\boldsymbol{\rho}) + \frac{\pi_s^2}{2M_s} + gk_b T \ln s \quad (3.5)$$

with π_s the additional degree of freedom and $g = 3N + 1$. Here π_s can be regarded as the virtual momentum of the heat reservoir and M_s is the corresponding effective mass. The equations of motion derived from the Hamiltonian in Equation (3.5) are then transformed back from the virtual systems leading to a set of equation of motions consistent to the canonical ensemble for a given temperature T . The described method was developed by Nosé [18], [19] and Hoover [20] and is therefore called the Nosé Hoover thermostat.

4. Atomic charges

Calculating electrostatic interactions within the framework of MD is only possible between localized charges (point charges). In a molecule on the other hand the negative charge is distributed over the entire molecule and therefore delocalized, while the nuclei with their positive charges are more or less on fixed positions. To overcome this the concept of atomic charges has been introduced, in which the basic idea is to map the negative charge from the electron cloud onto the nuclei. In this context one has to keep in mind that the atomic charge is just an auxiliary quantity for further calculations (e.g. MD) without any physical meaning.

There is a variety of schemes describing how to assign atomic charges that can be found in literature. They are designed to mimic some specific properties of a molecule.

One method is to span a grid over the molecule then assigning the charges to an atom by adding the charges of its nucleus and the grid points in its vicinity. In doing so the electrostatic potential of molecules can be fitted directly (electrostatic potential model (ESP)) [21] or with some modifications to ensure numerical stability (restrained electrostatic potential model (RESP)) [22]. The drawback of the methods presented above is that it is necessary to perform ab initio calculations in order to obtain the electron distribution prior to the actual fit process. For large molecules these ab initio calculations can be extremely time consuming.

In order to decrease the computational effort linked to the ESP method Jakalian et al. [23],[24] developed an algorithm managing to assign atomic charges with lower computational efforts, the so called AM1-BCC. In contrast to the ESP and RESP methods the AM1-BCC algorithm uses a computationally less time consuming but inaccurate procedure to obtain atomic charges, namely the AM1 method [25]. These AM1 charges are then modified by so called bond charge corrections (BCC). Basically, the BCC parameters are positive or negative charges depend on the bond type and on the involved atoms. They were derived by Jakalian et al. [24] for a large number of bond types and

4. Atomic charges

atoms and, therefore, do not have to be calculated anymore.

For the AMBER force field, which is used in this work, it is recommended to use the AM1-BCC method to derive the atomic charges, [6].

Part II.

Methodology

5. Overview

As already mentioned in the introduction, MD simulations were used to describe the dynamics of SAMs on a (111) surface of a gold substrate consisting of six atom layers. The gold atoms were held fixed at their lattice positions and the interactions among the gold atoms were turned off. Leaving the gold atoms at rest has been shown to be a reasonable approximation by Kim et al. [26].

The general AMBER FF [6] was used in order to simulate the interactions of the atoms within a molecule, its functional form is shown in Equation (2.1). The interactions between the individual molecules of the SAM were calculated by use of the non-bonded terms of the AMBER force field (last two terms in Equation (2.1)). To obtain the vdW parameters for interactions between atoms of different type (S - C, C - H and S - H) the mixing rules as described in Equation (2.8) and (2.9) were applied.

The interaction between the organic molecules and the gold was modeled using the vdW parameters of a modified version of the Dreiding FF [8], [27], the parameters are listed in Table 5.1. In the Dreiding FF, the vdW forces are described by means of the Buckingham potential similar to Equation (2.10)

$$u_{vdW} = D_{ij} \left[\left(\frac{6}{\zeta - 6} \right) e^{\zeta(1-\rho)} - \left(\frac{\zeta}{\zeta - 6} \right) \rho^{-6} \right] \quad (5.1)$$

with $\zeta = 12$ a constant and $\rho = \frac{R_{ij}}{r}$. Since the Buckingham potential was implemented in the MD code according to Equation (2.10) the parameters in Equation (5.1) had to be rewritten as follows

$$\begin{aligned} A_{ij} &= D_{ij} \left(\frac{6}{\zeta - 6} \right) e^{\zeta} \\ \rho_{ij} &= \frac{R_{ij}}{\zeta} \\ C_{ij} &= D_{ij} \left(\frac{\zeta}{\zeta - 6} \right) R_{ij}^6 \end{aligned} \quad (5.2)$$

Modeling the interactions of the gold atoms and the docking atoms as vdW like with a

5. Overview

Table 5.1.: *Parameters for the Buckingham potential 5.1. Although the parameters were modified according to Equation 5.2 they are listed in their original form as in the work of Jang et al. [27].*

D_{ij}	...	depth of the potential well	
r	...	distance at the energy minimum	
ζ	...	dimensionless scaling parameter	
	$D_{ij}[\text{kcal/mol}]$	$r[\text{\AA}]$	ζ
Au-S	9.003	2.682	12.0
Au-C	0.064	3.561	12.0
Au-H	0.041	3.082	12.0

deep potential well (see Table 5.1), allow for the docking groups and molecules respectively to hop from one docking site to another. This becomes particularly important when investigating low coverages.

The MD calculations themselves were performed using the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [28, 29] MD code. A typical simulation of thiols on a Au(111) surface consisted in principle of the following steps being adopted from the work of Dirama and Johnson [2]:

- At the beginning, each molecule was placed on its docking site (fcc-hollow) in an upright position and there randomly twisted around the z -axis. The potential energy of this configuration was then minimized by adjusting the atomic coordinates using the conjugate gradient method. A minimization is necessary because starting a run from a non-minimized configuration may cause an error when single atoms are too close to each other and thus the resulting forces happen to be too strong.
- The minimized system was then heated from 0 K to 900 K using the Berendsen thermostat (see Section 3.1.2). The time steps δt for all simulations, if not explicitly given, was set to 1 fs. During the heating process the docking atoms were restricted to move.
- At 900 K the system was equilibrated using the Nosé-Hoover thermostat (Section 3.1.4). Applying the Nosé-Hoover thermostat caused the system to move toward thermodynamical equilibrium and the molecules to be thermodynamically randomized. In order to thermalize only the internal coordinates (tilt angle, twist angle ...) the docking atoms were held fixed on their initial positions.

- After the equilibration at 900 K, the system was cooled to a certain temperature (e.g. 300 K) using the Berendsen thermostat, for further investigations. The rate of cooling the system is crucial, because on the one hand, cooling it too fast makes the system freeze in an unfavorable configuration in which it may be stuck, slower cooling rates, on the other hand, lead to higher computational costs.
- When cooled to a desired temperature, the system was again equilibrated at this temperature using the Nosé-Hoover thermostat. The equilibration by means of the Nosé-Hoover thermostat is necessary because the Berendsen thermostat does not sample a canonical ensemble [1], thus the system is not in thermodynamic equilibrium. The data to be analyzed was then obtained during a further equilibration without any thermostat representing a microcanonical ensemble. The time of equilibration usually depends on the temperature, i.e. the lower the temperatures the longer the equilibration times have to be.

6. Charge assignment

In order to take into account electrostatic interactions charges were assigned the atoms by means of the AM1-BCC scheme discussed in Chapter 4. By doing so several problems occurred due to the fact that the molecules used to build the monolayer were unsaturated, i.e., the hydrogen atom at the SH group was removed (see Figure 6.1 and 6.2).

To assign the charges the program *antechamber* from the AMBER toolbox [30] was used. Since *antechamber* is not capable of assigning charges to molecules with unusual valence states [31], the charges had to be assigned to the molecule with saturated thiol group (see Figure 6.1). For the purpose of modeling a monolayer on a Au(111) surface the hydrogen at the docking group had to be removed after the charge assignment. Investigations of SAMs with thiol docking groups by means of density functional theory (DFT) show that there is no notable charge transfer between SAM and substrate [32] and [33], thus no additional charges had to be assigned and the gold atoms were left uncharged.

The problem arising in this context is how to handle the charge assigned to the hydrogen atom at the thiol group, to tackle this the following two approaches were chosen:

1. One possibility is simply cutting off the hydrogen and leaving the residual molecule negatively charged. Since the interactions of the SAM molecules and the gold substrate are fully described by the vdW parameters and electrostatic interactions are not computed explicitly one might think of this as an appropriate approach but it has two significant drawbacks: First, the negatively charged molecules give rise to electrostatic interactions among of them which do not occur in a real SAM. Secondly, applying the P3M solver to calculate the long range electrostatic interactions to an overall negatively charged system (SAM + substrate) is problematic (see Section 2.2). To overcome this, the P3M routine introduces counter charges being equally distributed over the system which is in terms of physics questionable (see Figure 6.2a).
2. A physically more reasonable approximation is to add the charge of the cut off hydrogen to the sulfur atom resulting in an overall electrically neutral system. In

6. Charge assignment

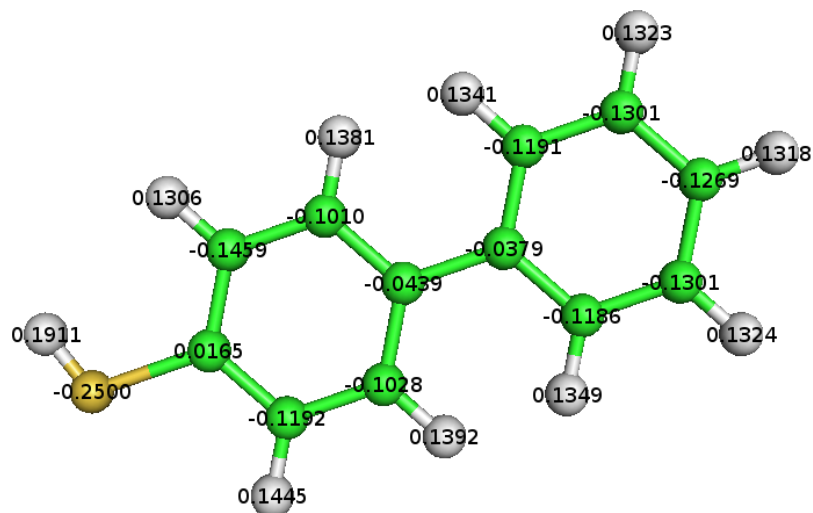


Figure 6.1.: *Saturated biphenylthiol molecule with AM1-BCC charges*

this case one can avoid computational problems related to the P3M method and additional electrostatic interactions among the SAM molecules do not occur (see Figure 6.2b).

The two methods discussed above are however just approximations to mimic the charge rearrangements occurring upon SAM formation since the charge distribution does not remain constant over time but changes as the position of individual atoms change.

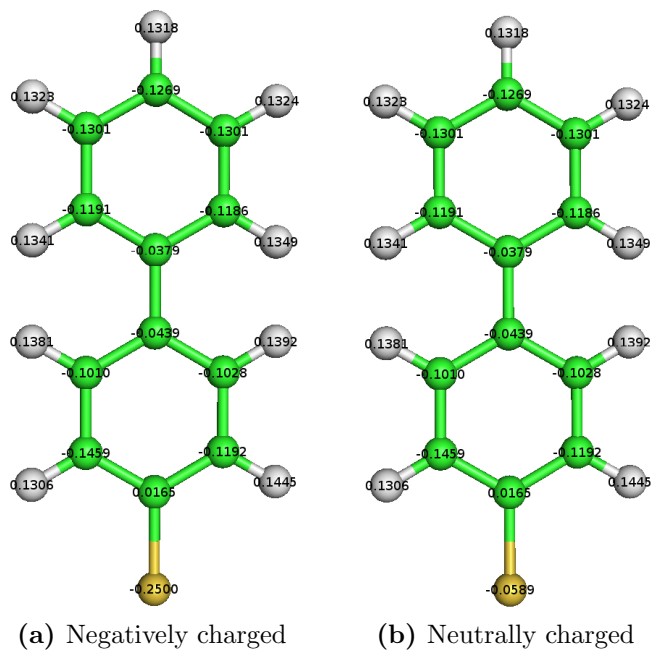


Figure 6.2.: *Unsaturated biphenylthiol molecules with atomic charges determined by the AM1-BCC method*

7. Input data

In this chapter the files needed to run a simulation using LAMMPS are presented and discussed. Further information can be found in Ref. [28]. In order to execute a calculation in LAMMPS basically two files are needed namely an *input file*, specifying the parameters of a simulation, and *data file* containing the atomic specifications (positions, masses, FF parameters). Strictly speaking, a 'data file' is not compulsory, one could quote the entries of the data file in the input file as well. However, for large systems it is more convenient to have atomic data and simulation data in two separated files.

7.1. The input file

A schematic input file is given in Appendix 1, the most important commands are described in the following section.

The units LAMMPS should use for calculations are determined via the command `units`. When specifying the FF parameters one has to pay attention if the values taken from literature are in appropriate units otherwise they must be converted. For all simulations in this work the `units` parameter was set to `real` by:

```
units real
```

Using `real` units means that, mass is given in grams/mole, distance in Angstroms, time in femtoseconds ..., a full description can be found in the LAMMPS online manual [28].

The boundary conditions were chosen to be periodic, this was done using the `boundary` keyword

```
boundary p p p
```

this means that the simulation box was replicated periodically in x , y and z direction.

7. Input data

The `atom_style` keyword defines the attributes assigned to the atoms, e.g. type, charge, position, molecule etc. For all calculations

```
atom_style full
```

was used, where `full` means that besides the type and the position of a particle also the atomic charge must be assigned.

The parameter `pair_style` specifies which pair potential is used for non bonded interactions. Since inter- and intramolecular interactions and interactions between SAM molecules and gold substrate were described by different FF the keyword `hybrid` had to be used:

```
pair_style hybrid buck 14.0 lj/charmm/coul/long 14.0 16.0
```

To model the interactions of SAM and gold substrate the Buckingham potential with a cut off radius of 14\AA (`buck 14`) was used, all other non bonded interactions were computed by means of a Lennard-Jones potential with an inner cut off radius of 14\AA and an outer one of 16\AA (`lj/charmm/coul/long 14.0 16.0`). Here the inner cut off determines up to what distance vdW interactions are taken into account, between inner and outer cut off the Lennard-Jones potential is multiplied by an additional switching term assuring that the potential energy smoothly approaches zero beyond the outer cut off.

Similar to the `pair_style` command the commands `bond_style`, `angle_style` and `dihedral_style` determine which FF for bond stretching, bond bending and dihedral is used respectively. These parameters were all set to `harmonic` since the AMBER FF is parameterized for harmonic potentials. It should be mentioned the functional form for describing the torsional energy is not really 'harmonic' (see Equation 2.1). When converting the FF parameters from the AMBER files to LAMMPS data files, the obtained parameters belong to the following potential function, defined by the keyword `harmonic`:

$$u_t = K[1 + \cos(n\phi)]. \quad (7.1)$$

A description how the LAMMPS data files were generated is given in the next section.

To take into account the influence of non bonded interactions on bonded atoms weight factors for 1,4 pairs were set by use of the line:

`special_bonds amber`

in the input script.

The electrostatic interactions were calculated using the P3M solver:

`kpspace_style pppm 1.0e-4`

The value `1.0e-4` determines the precession when assigning charges to the grid points of the P3M solver (see Section 2.2), the number of grid points is then derived by LAMMPS using the method in Ref. [15].

In order to generate a gold slab with a (111) surface the following command was used

```
lattice fcc 4.0782 orient x -1 -1 2 orient y 1 -1 0 orient z 1 1 1 &
spacing 1 1 1
```

By applying just the `fcc 4.0782` command, LAMMPS generated a simple fcc lattice with a lattice constant of 4.0782\AA resulting in a (100) surface. The `spacing 1 1 1` command defines that further declarations of distances, e.g, to define region sizes etc., are given in units of the lattice constant. To get a (111) surface the simple fcc lattice had to be reoriented by use of the following coordinate transformations illustrated in Figure 7.1. Suppose that the coordinate axes of the '(100)' system are x , y and z , the axes of the transformed system can be obtained by laying a plane through the the points

$$p_x = (1, 0, 0)$$

$$p_y = (0, 1, 0)$$

$$p_z = (0, 0, 1)$$

and constructing the auxiliary vectors

$$\mathbf{a} = \mathbf{p}_x - \mathbf{p}_y = (1, -1, 0) \quad (7.2)$$

$$\mathbf{b} = \mathbf{p}_z - \mathbf{p}_y = (0, -1, 1) \quad (7.3)$$

$$\mathbf{c} = \mathbf{b} \times \mathbf{a} = (1, 1, 1). \quad (7.4)$$

Here the vectors \mathbf{a} and \mathbf{c} are the new axes \mathbf{y}' and \mathbf{z}' of the transformed '(111)' system, since this system must be orthogonal the \mathbf{x}' axis can be calculated via

$$\mathbf{x}' = \mathbf{y}' \times \mathbf{z}' = (-1, -1, 2) \quad (7.5)$$

7. Input data

The resulting axes \mathbf{x}' , \mathbf{y}' and \mathbf{z}' were used in connection with the `orient` keyword to generate a lattice with (111) planes as its horizontal planes.

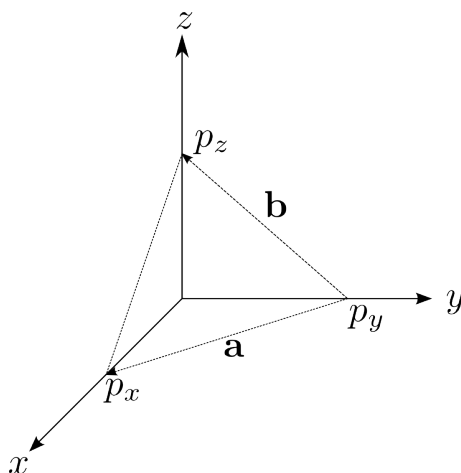


Figure 7.1.: *Auxiliary vectors used in the coordinate transformation*

The atomic data of the monolayer was included by reading a data file via the `read_data name_of_data_file`

command. The data file contains atomic parameters, such as position, charge, mass, type etc., FF parameters and the borders of the simulation box

The gold slab was created in a block shaped region of the width and length of the simulation box and of the height providing space for six atomic layers by use of the lines

```
region au block INF INF INF INF -3.8 -0.5
create_atoms 5 region au
```

where `au` is the name of the region and `5` is the atom type of the gold atoms. In other words, these commands generate a region within the simulation box filled with atoms of type `5` at the lattice sites of the lattice defined above.

The parameters for the interaction with the gold atoms were set by the command

```
pair_coeff 1 5 buck 1.5e6 0.22 6723.8
```

in this example the coefficients for the Buckingham potential between atoms of type `1` and type `5` are defined. The numbers `1.5e6 0.22 6723.8` refer to the parameters A_{ij} ,

ρ_{ij} and C_{ij} in Equation 2.10. Since the Coulomb interactions between gold atoms and the molecules had already been taken into account in those parameters no additional Coulomb terms in the the Buckingham pair potential were necessary.

The commands

```
neighbor 2.0 bin
neigh_modify every 1 delay 0 check yes exclude type 5 5
```

define parameters connected to building the neighbor list. The line `neighbor 2.0 bin` specifies the 'skin distance' of 2 Angstroms and the method used to build the neighbor list. The skin distance is added to the force cutoff distances and particles within this distance (cutoff + skin) are stored in the neighbor list. A detailed description of the skin parameter and neighbor lists can be found in the LAMMPS online manual [28] and in Chapter 3.4 of Ref. [10].

The arguments `every 1 delay 0 check yes` of the `neigh_modify` command determine how often a neighbor list is built. The interaction among atoms of type 5 (gold - gold) is turned off by the keyword `exclude`.

After having set these fundamental parameters one can in principle launch a simulation using the `run` command, further commands depend on the particular problem.

7.2. Setting up the data file

In this section a brief description of how the data files were generated is given. In Appendix 2 an example of a data file is shown.

Before describing how a data file was created the principle structure of such a file will be discussed.

At the beginning of a data file the number of atoms, bonds, angles and dihedrals are given. After that, number of different atom types, bond types, angle types and dihedral types used in the simulation are stated. Finally, the dimensions of the central simulation cell are specified.

Subsequent to these entries, there are the following sections:

Masses In this part the masses of the individual atom types are defined.

7. Input data

Pair Coeffs Here the pair coefficients for the non-bonded interactions are listed.

Bond Coeffs This sections contains the parameters for the bonded interactions.

Angle Coeffs List of the angle-bending coefficients.

Dihedral Coeffs The dihedral coefficients are given here.

Atoms Here all the atoms of the monolayer are listed. Since the gold atoms were created via the `create` command in the input file, they are not stated here.

Bonds In this part all the bondings within the SAM are defined. That is, to all bonded atoms a bond type is assigned.

Angles Similar to the 'Bonds' section the angle-stretching parameters are assigned to the bonded atoms.

Dihedrals The dihedral types are assigned to the respective atoms.

In order to create a data file described above, the molecular structure of a single saturated SAM molecule must be created. This was done using the program MOLDEN [34]. After that, charges were assigned to the atoms of the molecule according to the AM1-BCC scheme. The resulting file was then modified in the *leap* program of the AMBER toolbox, i.e. the hydrogen at the thiol docking group was removed. Eventually the modified molecule data file was saved as an AMBER topology and coordinate file, with the program *amber2lammmps*, coming with the LAMMPS package, a LAMMPS data file was generated from the AMBER files. In order to obtain a data file for a whole monolayer the self-written script *build layer* was used. This script reads in a data file for a single molecule and replicates the positions of the atoms in x and y direction. The molecules of resulting monolayer are then arranged in a $(\sqrt{3} \times \sqrt{3}) - R30^\circ$ surface lattice. Additionally the molecules are twisted randomly around their molecular axis. The source code of the *build layer* script is given in Appendix 3.

Further modifications were necessary in order to include the gold atoms, for this purpose the number of atom types in the 'Atoms' section of the data file had to be increased by one, see Appendix 2. Furthermore, the mass of the gold atom had to be defined in the 'Masses' section and the pair coefficients for the gold atoms were set in the section 'Pair Coeffs'. The values for the Lennard-Jones potential for gold were set to some dummy numbers since the pair coefficients for the gold - molecule interactions were explicitly specified in the input file and the gold-gold interaction was turned off.

A snapshot of a simulated system is depicted in Figure 7.2.

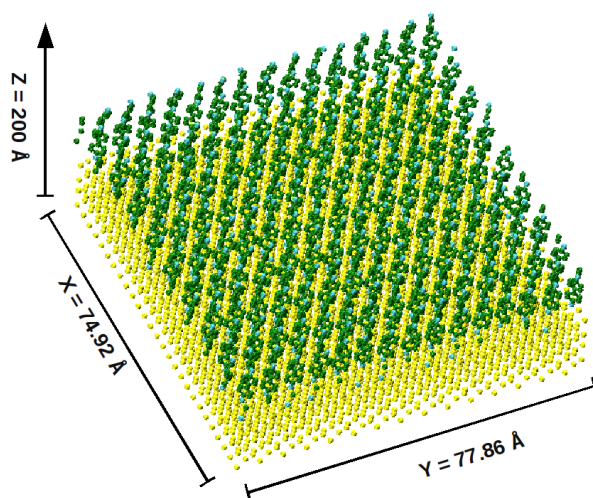


Figure 7.2.: Snapshot of a gold-biphenyldithiol system with 15×18 molecules arranged in a $(\sqrt{3} \times \sqrt{3}) - R30^\circ$ lattice

8. Data evaluation

8.1. Angle distributions

This section describes how the calculated angles were extracted from the MD output files and how subsequently angle distributions were calculated. The coordinates of the vectors needed to calculate the angles were obtained from the atomic coordinates stored in LAMMPS output files. In these files the atomic coordinates of all SAM atoms were dumped every 1000 steps during the NVE equilibration using the `dump` command (see Appendix 1). Further data evaluations were then done in MATLAB.

8.1.1. Tilt angle

The tilt angle Θ denotes the angle between the molecular axis and the z - axis, it is schematically depicted in Figure 8.1. Since the molecules were not totally rigid and flat the molecular axis was determined by averaging over several connecting vectors, i.e. the connecting vector between the docking atom and the top carbon atom and between the lowest and the highest carbon atom.

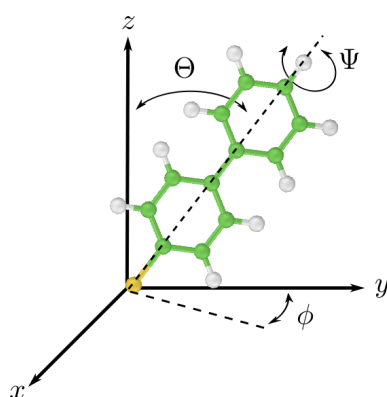


Figure 8.1.: Schematic representation of the investigated angles, due the work of [2]

8.1.2. Twist angle

The twist angle Ψ is determined by the plane of the phenyl rings and the $y - z$ - plane (see Figure 8.1), the angle itself was computed by means of the plane normals of the respective ring planes. Due to the fact that the two phenyl rings were subtly twisted with respect to each other, averages of the ring's plane normals were used for calculations.

8.1.3. Azimuthal angle

The angle between the plane spanned by the molecular axis and the z - axis and the $y - z$ - plane is called azimuthal angle ϕ , since this is an angle between two planes it was determined by means of surface normals similar to twist angle. An equivalent definition is to define the azimuthal angle as the angle between the projection of the molecular axis on the $x - y$ plane and the y axis.

After having calculated the angles described above, distributions of those angles were created. In doing so, histograms of the angles were computed for each time step when data were recorded. The resolution for those histograms was set to 0.1° for the tilt angle and 0.5° for twist and azimuthal angle. The final distributions were eventually obtained by summing up all the histograms and normalizing them. In this context, normalizing means that the sum over all histogram entries equals one. Finally, the data of the histograms was fitted by a least-squares fit using a Gaussian function

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}, \quad (8.1)$$

as fit function. The fit parameters b and c in Equation (8.1) denote the expectation value and the standard deviation. The parameter a , indicates the height of the peak, is related to c via

$$a = \frac{1}{c\sqrt{2\pi}}. \quad (8.2)$$

For distributions with a single peak (tilt and azimuthal angle) using a as fit parameter is redundant due to Equation (8.2). This is somewhat different in cases of two Gaussian like peaks as in the twist angle distribution here it is necessary to define a as an independent parameter in order to ensure that the overall area underneath both peaks equals one. The source code of the MATLAB script, used to obtain the described data, is given in Appendix 4.1.

8.2. Cumulative averages

In order to ensure that the system is in equilibrium the cumulative averages of the angles described in the previous section were observed. The cumulative average of a quantity X up to certain time step t_i is defined by

$$\bar{X}_i = \frac{X_0 + X_1 + \dots + X_i}{i}. \quad (8.3)$$

In equilibrium the cumulative average should converge to some constant value in time.

8.3. Mean square displacement

The MSD describes the square of the relative change in position with respect to the time t_0 averaged over all considered particles, in this work the docking atoms. The according formula is given in Equation (8.4).

$$\langle u(t) \rangle = \langle [r(t + t_0) - r(t_0)]^2 \rangle \quad (8.4)$$

The MSD was obtained by applying the `fix msd` command in LAMMPS ¹.

¹In later version of LAMMPS the command `fix msd` has been replaced by `compute msd`

8.4. Displacements

Furthermore, a histogram of the displacements of all docking atoms was evaluated. In order to do so, the displacements of the docking atoms with respect to their initial position (i.e. at the beginning of an equilibration) were computed and then depicted in a histogram with steps of 0.05\AA . In principal one can read from those histograms how many docking atoms moved how far. For a Au(111) surface, the distances between the docking sites are given in Table 8.1 and illustrated in Figure 8.2, respectively.

Additional to the distribution of the displacements the fraction of the docking atoms located at each kind of docking site was calculated and represented in a histogram, the cutoff radius for assigning a docking group to docking site was chosen to be 0.83\AA corresponding to the radii of the circles in Figure 8.2.

When considering the first three layers of a slab with a (111) surface, there are four kind of different docking sites, namely fcc-hollow, hcp-hollow, top and bridge site. The hcp-hollow site is located above the atoms of the deepest layer, the hcp-hollow site above the atoms of the middle layer, and the top sites above the atoms of the top layer. The bridge sites is a transition state located in between two atoms of the top layer. The docking sites are illustrated in Figure 8.2.

The MATLAB script used for the displacement analysis is shown in Appendix 4.2. In order to allocate the docking atoms to a certain docking site the following procedure was used:

1. The coordinates of the fcc-hollow, hcp-hollow, bridge and top sites were generated and stored in a list.
2. The distances between the docking atom at a certain step and the fcc-hollow sites were determined for every individual molecule. If the the distance was smaller than the cutoff radius of 0.83\AA , the docking site was associated to fcc-hollow. In the case that the distances were larger, this procedure was repeated for hcp-hollow and top sites.
3. If none of the criteria of step two were met, the docking atom was associated with a bridge site.

Table 8.1.: Distances between docking sites on $Au(111)$ surface based on primitive lattice constant of 4.0782 \AA ; all values are given in \AA

	fcc-hollow	hcp-hollow	top	bridge
fcc-hollow	-	1.66	1.66	0.83
hcp-hollow	1.66	-	1.66	0.83
top	1.66	1.66	-	0.83
bridge	0.83	0.83	0.83	-

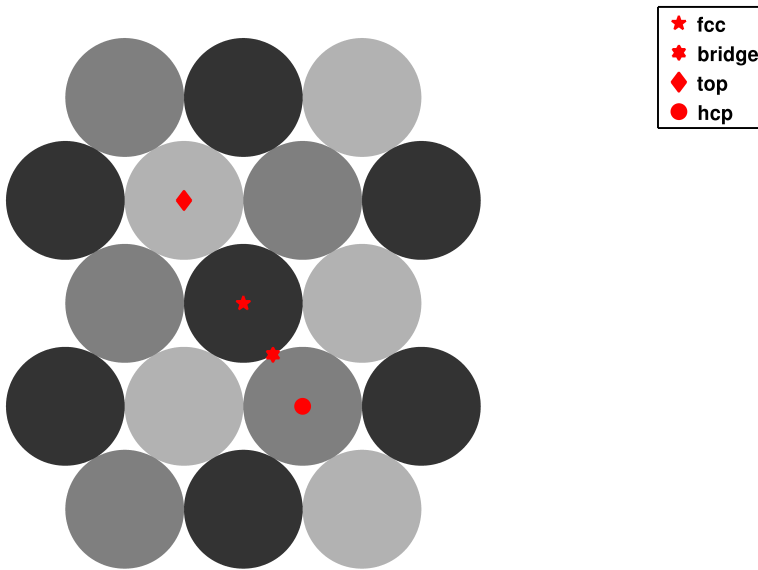


Figure 8.2.: Graphical representation of the docking sites on a $Au(111)$ surface. The light circles correspond to the top layer atoms, the dark to the lowest ones.

Part III.

Results and discussion

9. Full coverage systems

9.1. Biphenyldithiol SAM on a Au(111) surface

The first system to be investigated was Biphenyldithiol on an Au(111) surface, the Biphenyldithiol molecule with its AM1-BCC charges is shown in Figure 9.1. The molecules were arranged in the manner described in Chapter 7 with 15 molecules in x and 18 in y direction resulting in a simulation cell of $74.92 \times 77.86 \times 200$ Å. The goal of investigating Biphenyldithiol on gold was to reproduce the data from Dirama and Johnson, [2] at 300 K and to get familiar with the MD package. In order to get a grip on the LAMMPS code further investigations at 200K, 150K, and 100K were made where at each temperature angle distributions were calculated and a displacement analysis was performed.

9.1.1. Investigations at 300 K

In order to reproduce the data of Dirama and Johnson, [2] the parameters, if accessible, were taken from their work, in Table 9.1 these parameters are listed. Besides the brief annotations of the single quantities more detailed descriptions can be found in the LAMMPS manual [28]. The tolerance values for the initial minimization (E_{tol} and F_{tol}) were chosen to be quite small, but since the initial minimization has more technical reasons, those parameters have negligible influence on later results. The value of the damp factor used in the Berendsen thermostat ($T_{damp}^{Berendsen}$) was selected in order to get a linear change in temperature (see Figure 9.9). During NVT equilibration, the damp factor for the Nosé-Hoover thermostat was chosen in such fashion, that the temperature fluctuations were small. The angle distributions of these calculations are illustrated in the Figures 9.2 9.3 and 9.4; the respective fit parameters are listed in Table 9.2.

The agreement of my results with those of Dirama and Johnson is fairly good (see Table 9.2), the only divergence is in the distribution of the azimuthal angle. The expectation

Table 9.1.: *Parameters for MD simulations of Biphenyldithiol on gold, values with an asterisk were taken from [2]*

E_{tol} [Kcal/mol]	...	Energy tolerance during the initial minimization of the system
F_{tol} [Kcal/(moleAngstrom)]	...	Force tolerance during the initial minimization of the system
$T_{damp}^{Berendsen}$ [fs]	...	Damp factor of the Berendsen thermostat
T_{damp}^{NVT} [fs]	...	Damp factor for the Nosé - Hoover thermostat used for NVT equilibration
t_{900}^{NVT} [ps]	...	Equilibration time of the NVT equilibration at 900 K
t_{300}^{NVT} [ps]	...	Equilibration time of the NVT equilibration at 300 K
t_{300}^{NVE} [ps]	...	Equilibration time of the NVE equilibration at 300 K

E_{tol}	F_{tol}	$T_{damp}^{Berendsen}$	T_{damp}^{NVT}	t_{900}^{NVT}	t_{300}^{NVT}	t_{300}^{NVE}
$1e-8$	$1e-8$	1	10	400*	50*	100*

Table 9.2.: *Fit parameters of the angle distributions of Biphenyldithiol on gold at 300 K, the values with an asterisk are the results of Dirama and Johnson [2]*

μ [degree]	...	Expectation value of the tilt, twist and azimuthal angle
σ [degree]	...	standard deviation of the tilt, twist and azimuthal angle

Angle	μ	σ	μ^*	σ^*
Tilt	13.2	2.4	13.0	2.6
Twist	51.4	8.5	52.5	8.7
	128.1	8.5	127.5	8.6
Azimuthal	179.7	14.1	-1.5	3.3

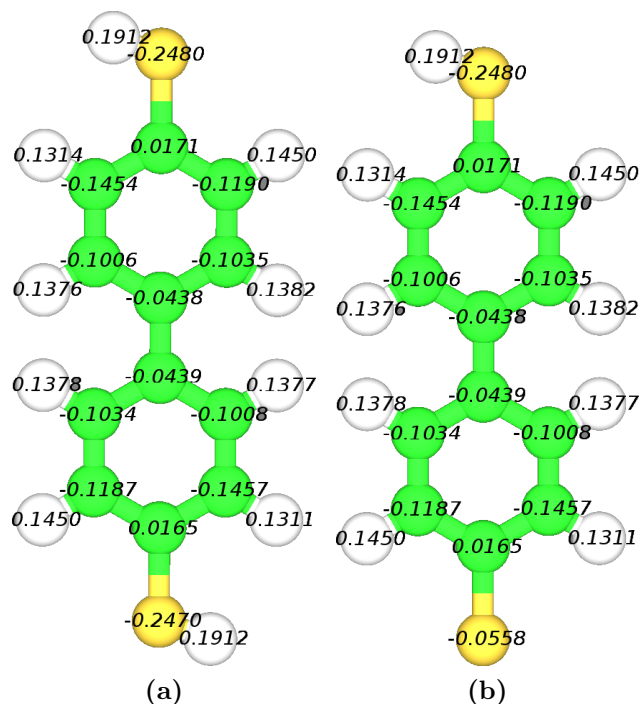


Figure 9.1.: Saturated Biphenyldithiol (a) and with cut off hydrogen at the docking site (b). The partial charges were derived using the AM1-BCC assignment scheme

value for the azimuthal angle seems just rotated around roughly 180 degrees, whereas its standard deviation differs significantly. The reason for this divergence may stem from different values of the assigned atomic charges.

In Figure 9.5 the percentage docking atoms at each kind of docking sites is shown, the majority of the docking groups stay at *fcc* sites for all timesteps indicating that the docking groups hardly move. The fact that the docking groups hardly move is emphasized in Figure 9.6 where the distribution of the displacements during the NVE equilibration is depicted. It can be seen that the docking groups remain in the vicinity of their initial docking sites (*fcc*). In a fully covered system this behavior appears to be expectable because the molecules are restricted from moving freely due to the lack of vacant docking sites.

Another interesting feature shows Figure 9.7 in which the averaged angles over over all molecules at single timesteps are plotted versus time. Whereas the twist angles remain at a constant value over time, there are slight oscillations in the case of the tilt angle and

9. Full coverage systems

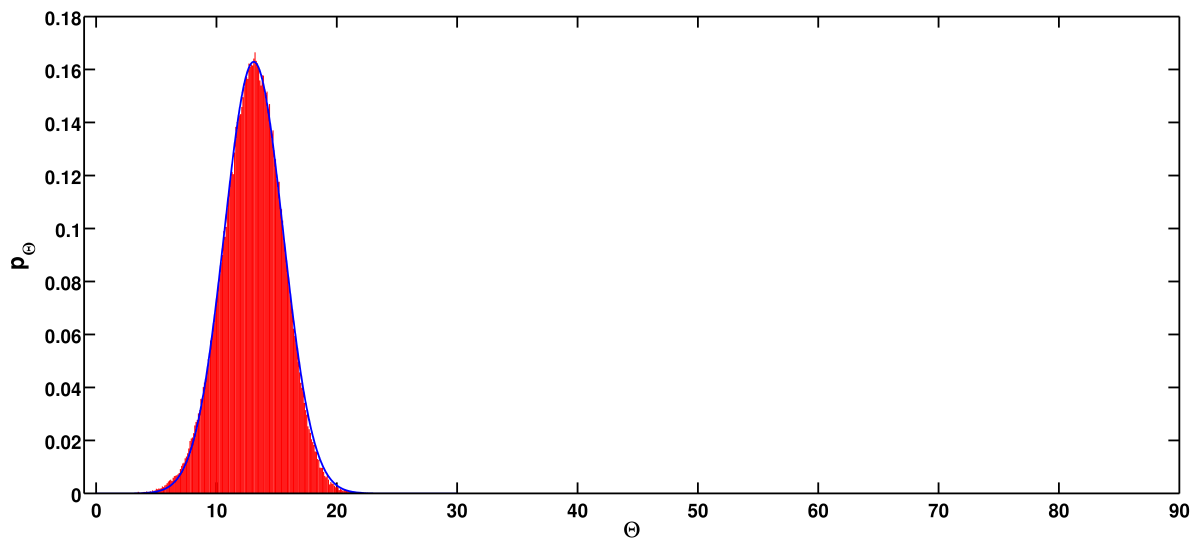


Figure 9.2.: Distribution of the tilt angle Θ recorded during equilibration in the NVE ensemble for 100 ps at 300 K

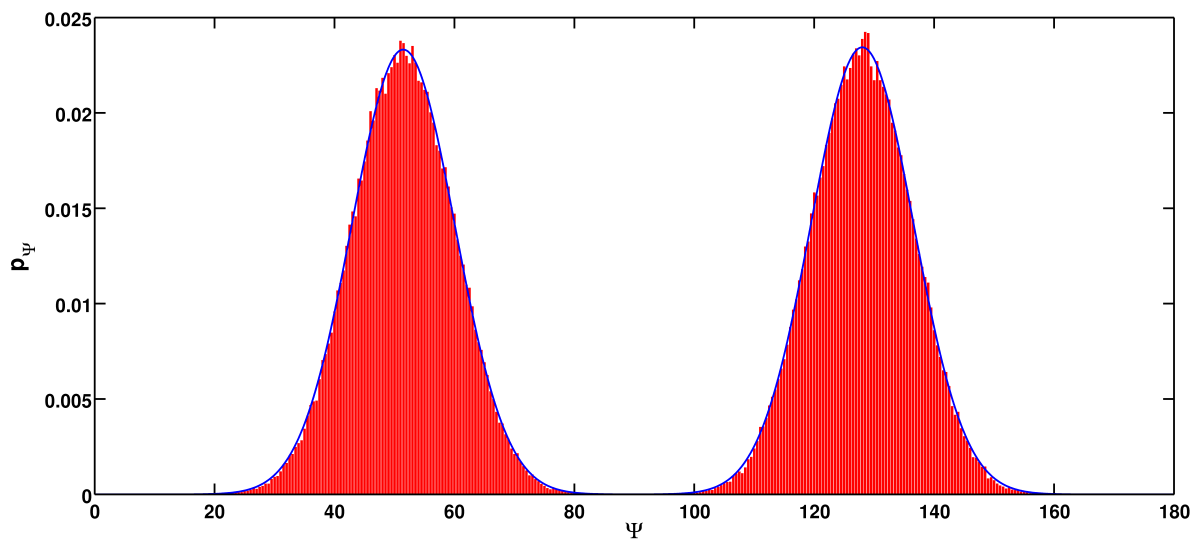


Figure 9.3.: Distribution of the twist angle Ψ recorded during equilibration in the NVE ensemble for 100 ps at 300 K

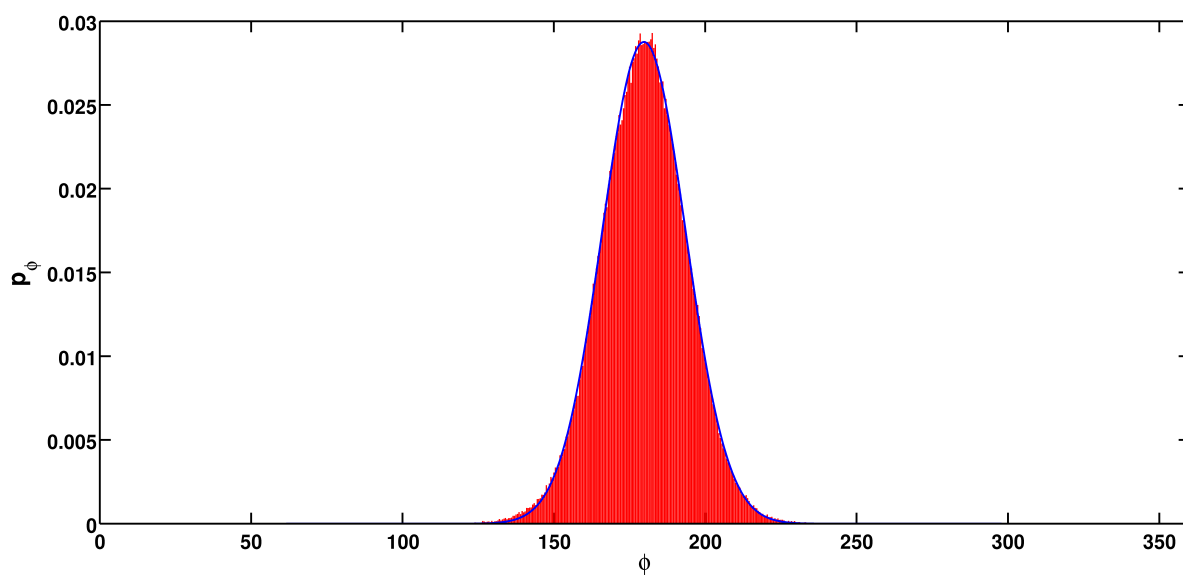


Figure 9.4.: *Distribution of the azimuthal angle ϕ recorded during equilibration in the NVE ensemble for 100 ps at 300 K*

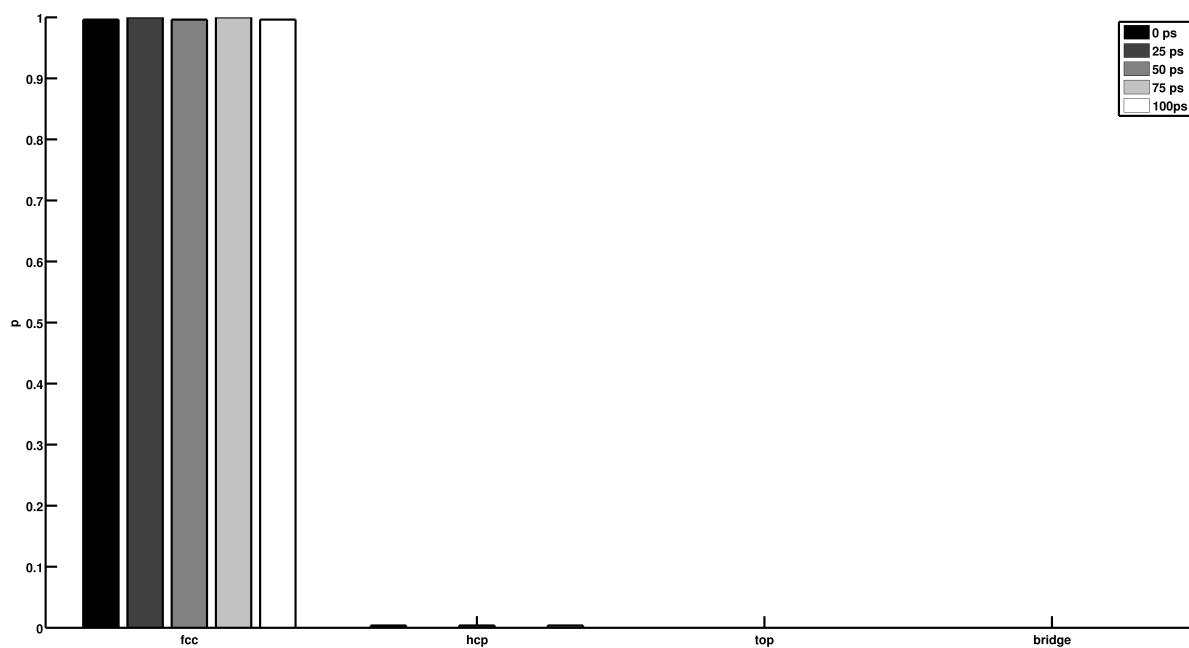


Figure 9.5.: *Distribution of occupied docking sites of certain timesteps during NVE equilibration at 300 K.*

9. Full coverage systems

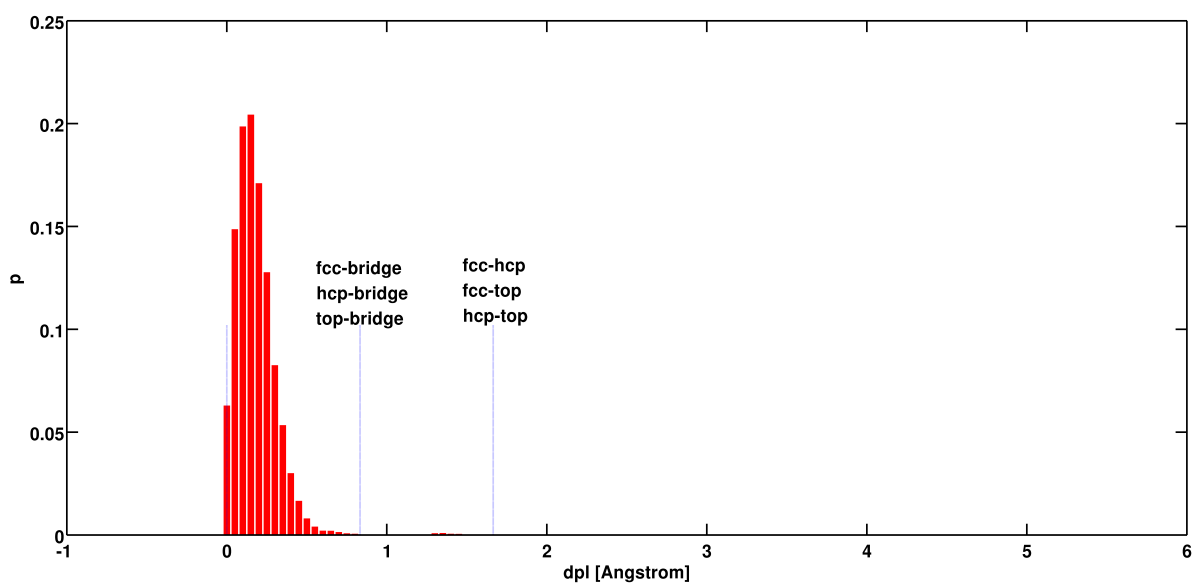


Figure 9.6.: Histogram of the displacement of the docking atoms during NVE equilibration at 300 K

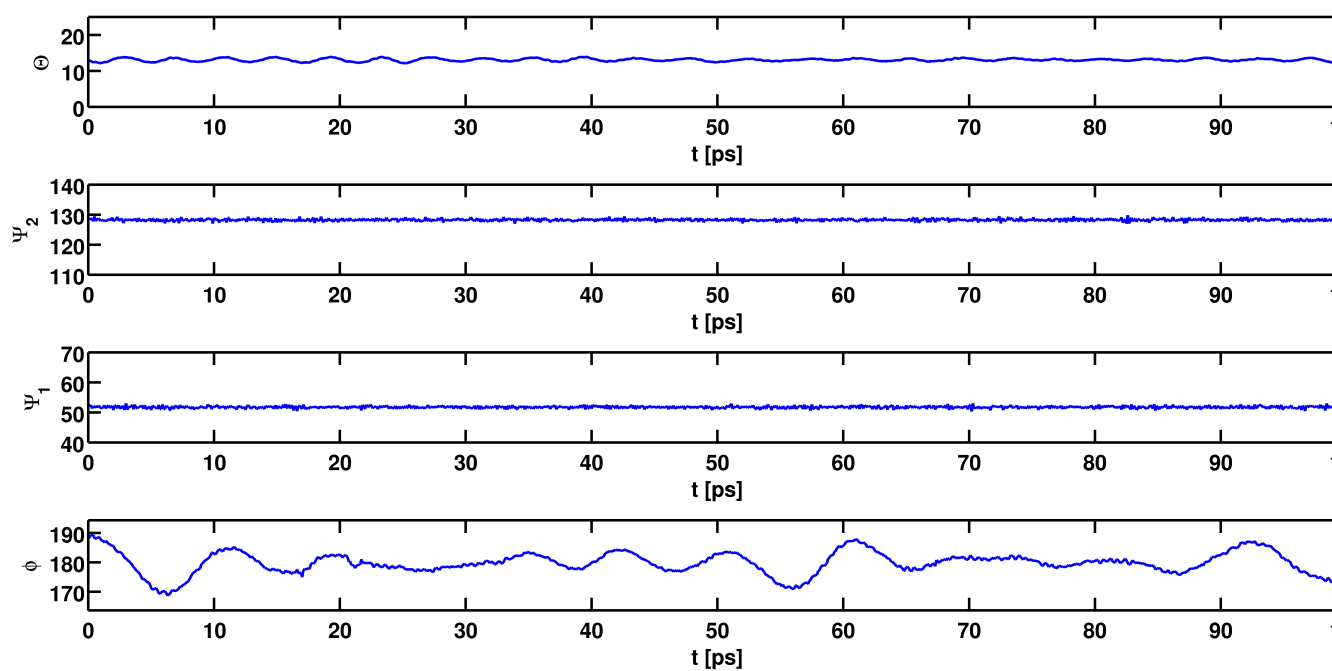


Figure 9.7.: Averaged angles versus time during NVE equilibration at 300 K

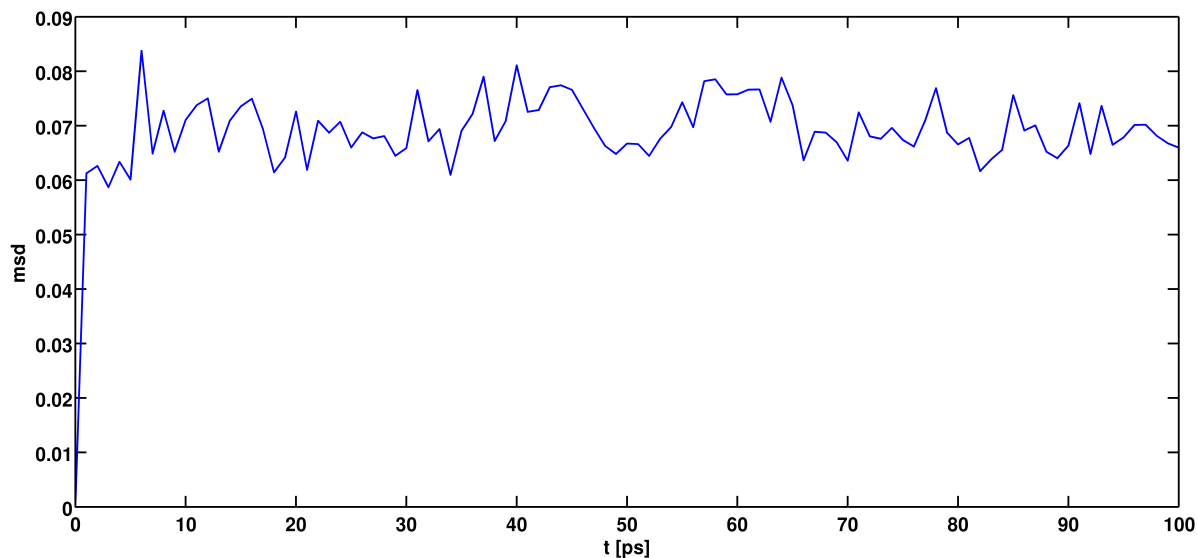


Figure 9.8.: *Mean square displacement of the docking atoms during NVE equilibration at 300 K*

strong variations for the azimuthal angle. The oscillating values for tilt and azimuthal angle indicate that tilting and rotating the molecules with respect to the z axis seems to be a fairly soft degree of freedom.

Finally the mean square displacement was calculated. In Figure 9.8 the MSD is plotted over the time, one can see that it quickly reaches a value around 0.7 \AA^2 and oscillates around that value which is in good agreement with the results of Dirama and Johnson [2].

9.1.2. Further investigations

In this section, results of simulations at lower temperatures than 300 K, namely at 200 K, 150 K, and 100 K are presented. Like in the section above, distributions of the tilt, twist and azimuthal angle and the MSD at the mentioned temperatures were calculated. An overview of the evolution of the temperatures and the total energy is given in Figure 9.9, as a cooling rate 0.002 K/step was chosen. The equilibration times at certain temperatures are listed in Table 9.3, at lower temperatures the equilibration times were chosen to be longer. In Figure 9.10 and 9.11 the cumulative averages of the individual angles at different temperatures are depicted. It can be found that, at lower temperatures it takes the cumulative averages longer to reach a constant value giving rise to longer equilibration times.

9. Full coverage systems

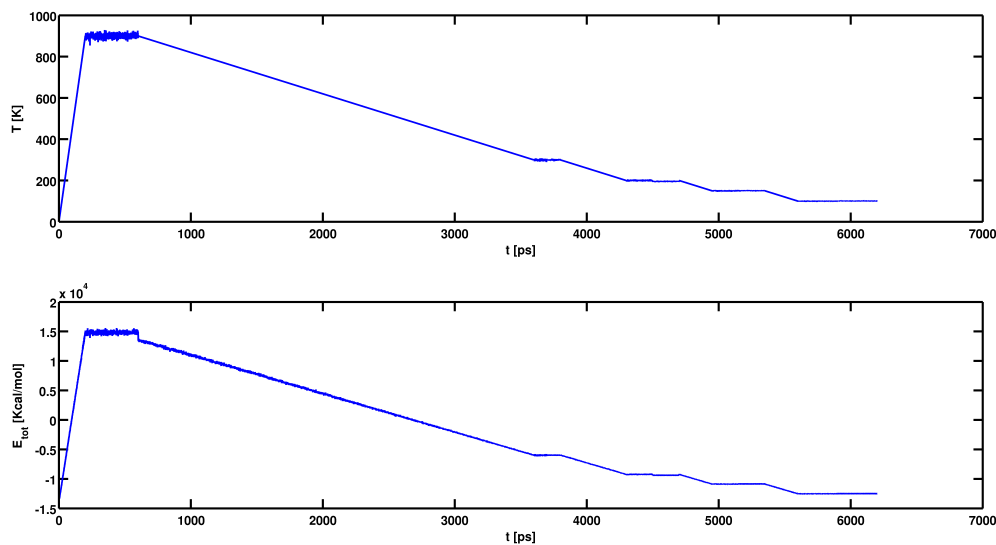


Figure 9.9.: *Developing of the temperature (upper graph) and the total energy E_{toto} (lower graph) during a simulation.*

Table 9.3.: *Equilibration times for the NVT and NVE equilibration at different temperatures*

T	Equilibration times	
	NVE	NVT
300 K	100 ps	100 ps
200 K	200 ps	200 ps
150 K	200 ps	200 ps
100 K	300 ps	300 ps

In Figure 9.12 the distributions for the tilt, twist and azimuthal angle are shown. A narrowing of the distributions with lower temperatures can be observed for all angles, which is consistent with thermodynamical considerations that the system ought to be more ordered at lower temperatures due to the lower kinetic energy. In addition to the narrowing of the distribution of the tilt angle, resulting in lower standard deviations, there is also a notable shift of the expectation value (peak position) of this angle. Figure 9.13 illustrates the decrease of the distributions' peak widths at lower temperatures. The expectation values and standard deviations at different temperatures are summarized in Table 9.4.

9.1. Biphenyldithiol SAM on a Au(111) surface

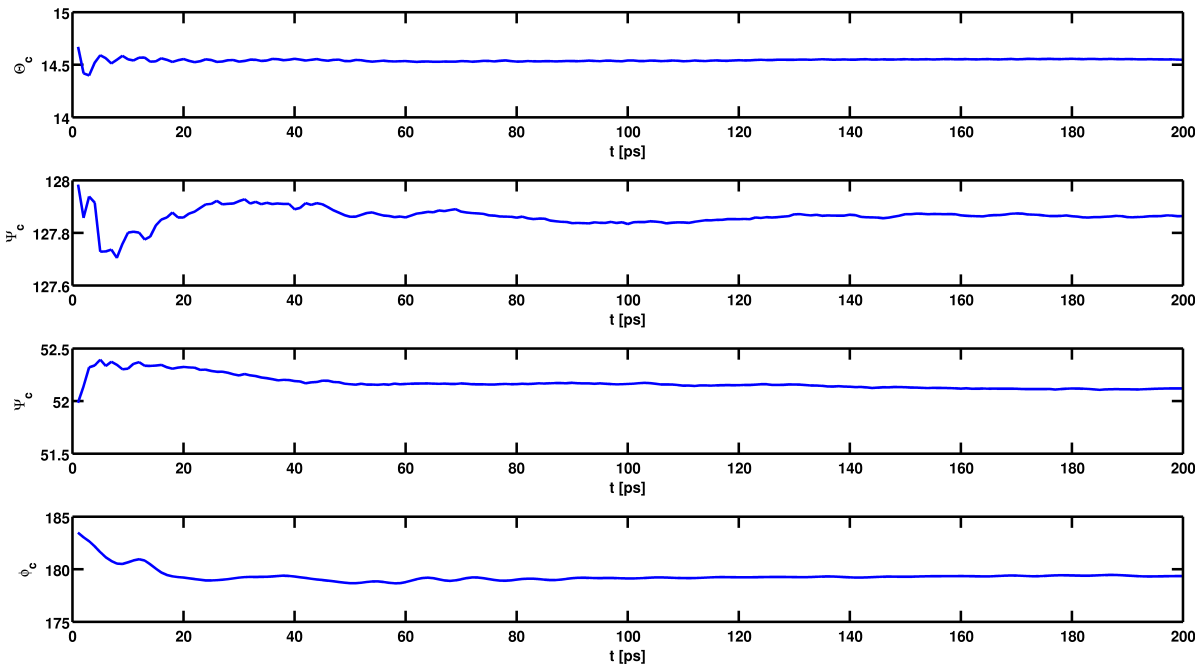
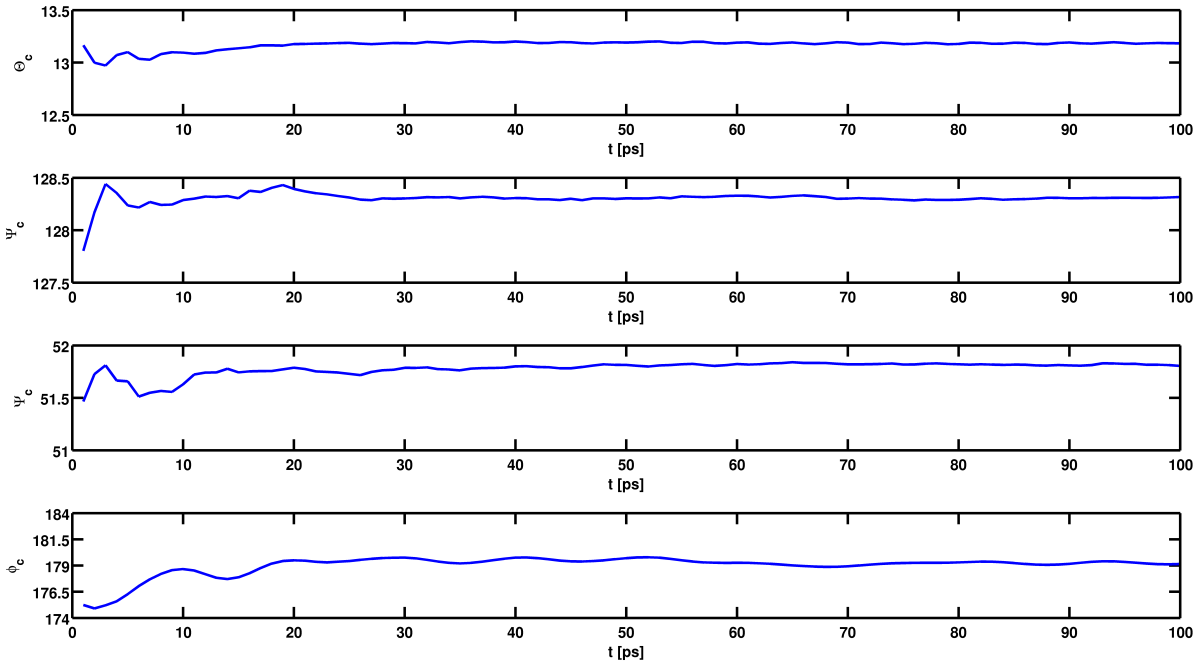
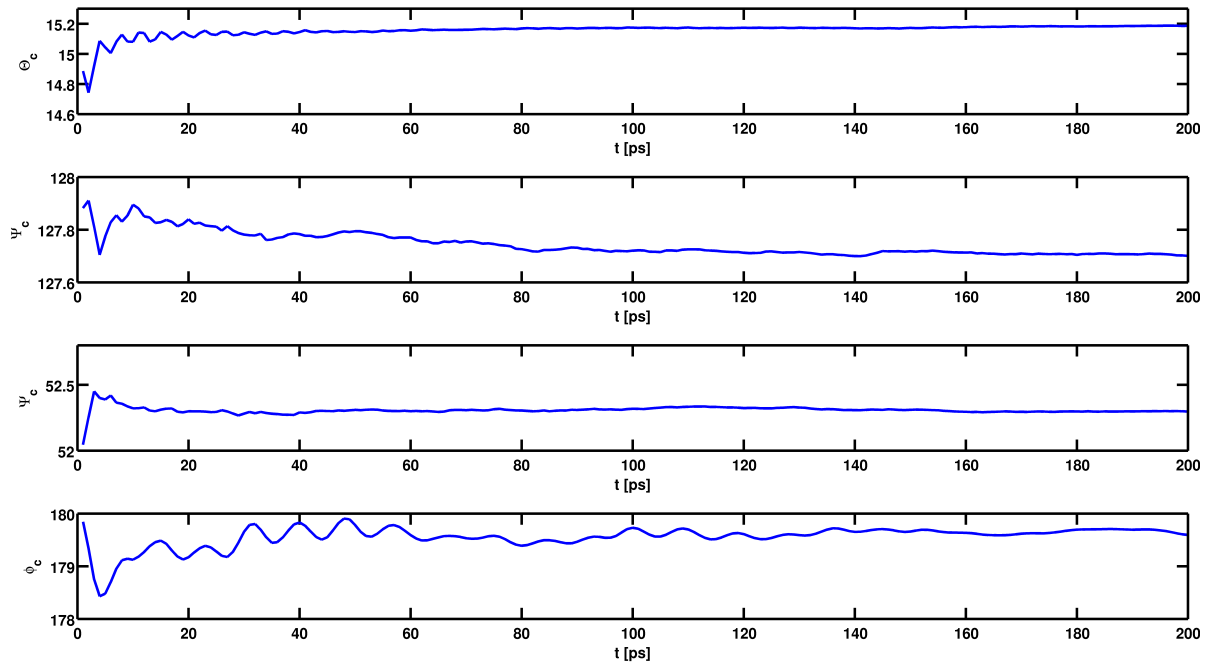
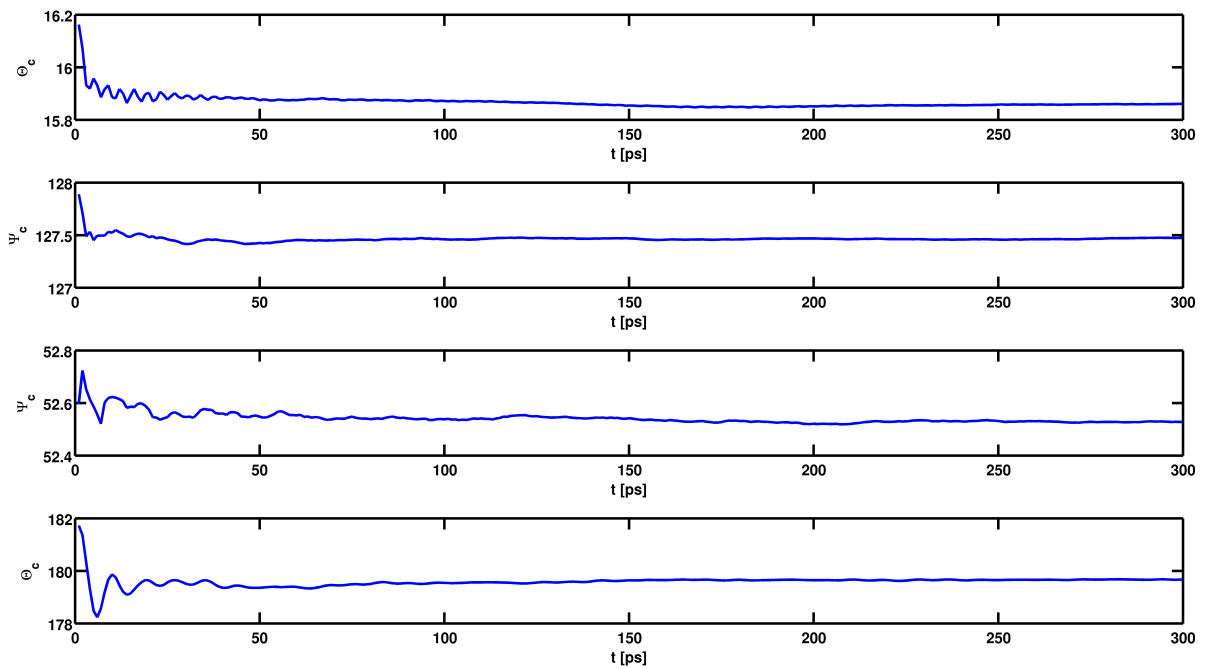


Figure 9.10.: Cumulative averages of the tilt (Θ), twist (Ψ) and azimuthal (ϕ) angle at 300 K and 200 K

9. Full coverage systems



(a) 150 K



(b) 100 K

Figure 9.11.: Cumulative averages of the tilt (Θ), twist (Ψ) and azimuthal (ϕ) angle at 150 K and 100 K

9.1. Biphenyldithiol SAM on a Au(111) surface

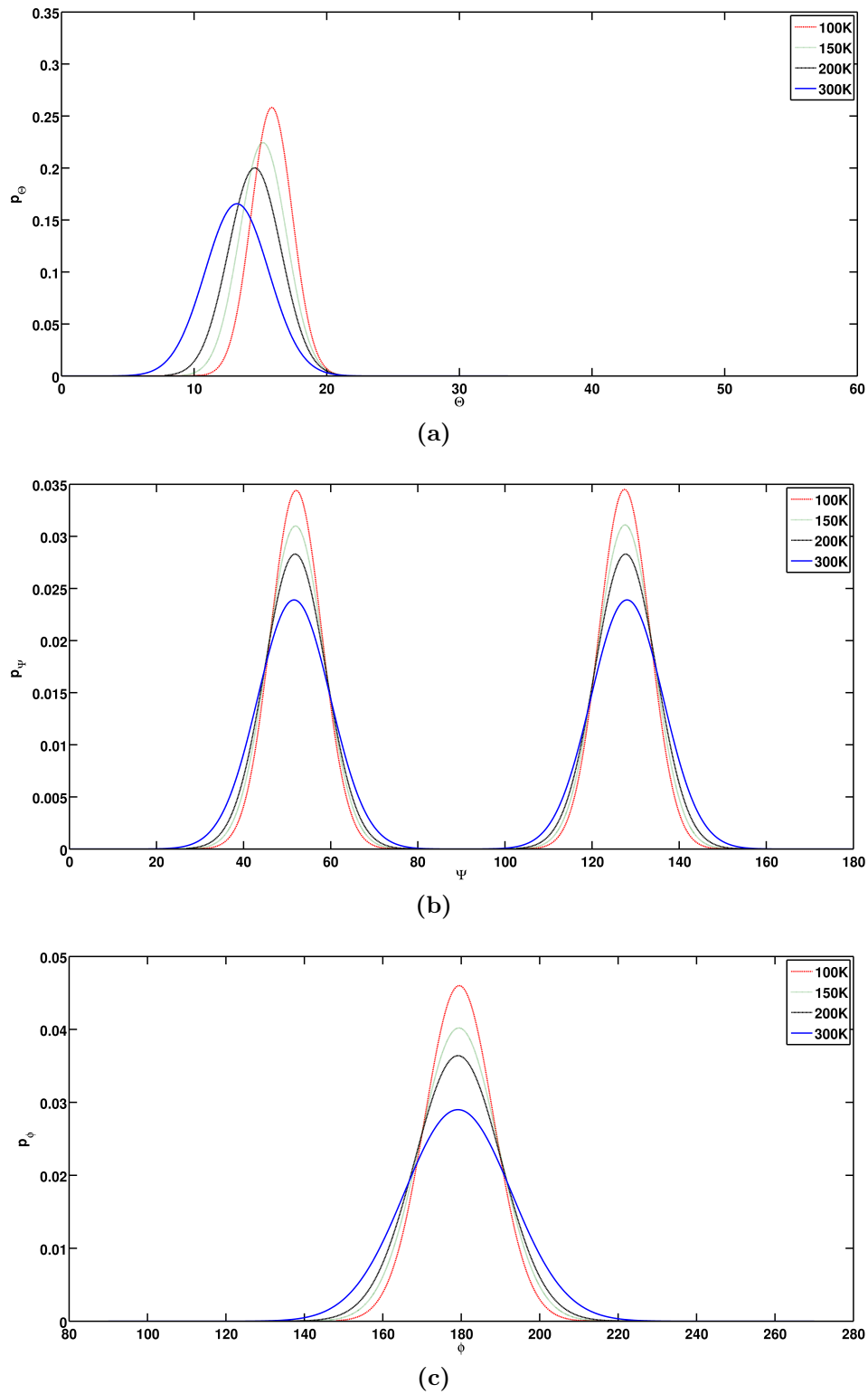


Figure 9.12.: Distributions of the (a) tilt, (b) twist and (c) azimuthal angle of biphenyldithiol for different temperatures

9. Full coverage systems

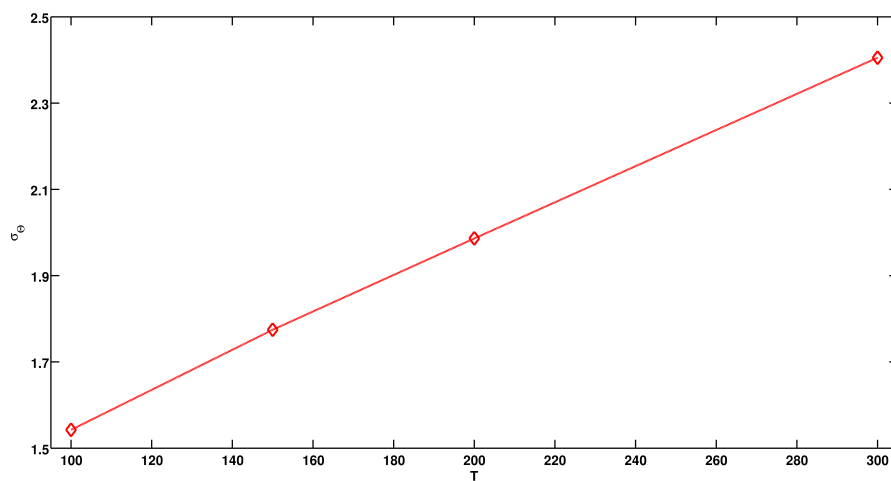
Table 9.4.: *Expectation values and standard deviations of the angle distributions of biphenyldithiol at 300 K, 200 K, 150 K and 100 K*

	Θ	...	tilt angle		Ψ	...	twist angle		ϕ	...	azimuthal angle		σ	...	standard deviation
$\mathbf{T}[\text{K}]$	Θ	σ_{Θ}	Ψ_1	σ_{Ψ_1}	Ψ_2	σ_{Ψ_2}	ϕ	σ_{ϕ}							
300	13.2	2.4	51.4	8.5	128.1	8.5	179.7	14.1							
200	14.6	2.0	51.8	7.1	127.7	7.1	179.2	11.0							
150	15.2	1.8	52.0	6.5	127.6	6.4	179.4	10.0							
100	15.9	1.5	52.1	5.9	127.4	5.8	179.5	8.8							

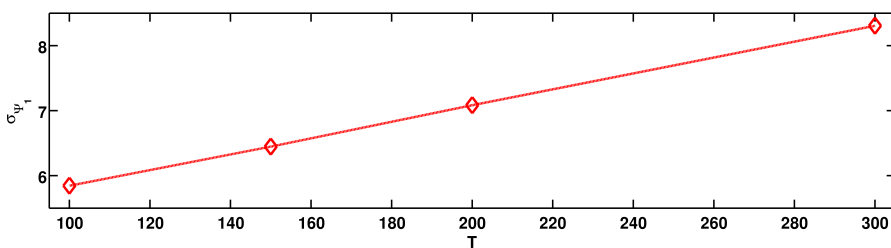
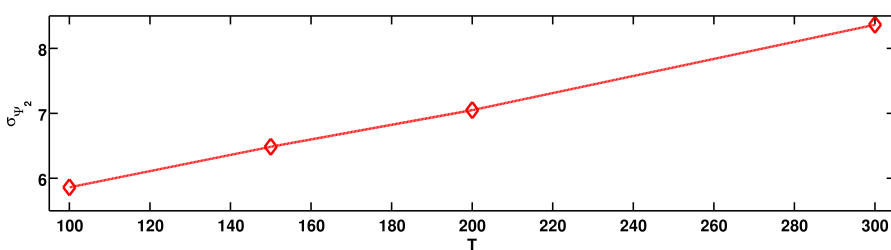
The MSD of the docking atoms at different temperatures is depicted in Figure 9.14 where a significant drop and smaller oscillations of the MSD can be observed with decreasing temperatures. Going to lower temperatures is related to lower kinetic energies and the docking groups do not tend to move so much anymore, thus leading to a lower MSD for lower temperatures.

Finally, a top view snapshot of the simulation cell during NVE equilibration at different temperatures is shown in Figure 9.15. One major feature of Figure 9.15 is the herringbone structure of the SAM molecules also indicated by the two peaks in the twist angle distribution in Figure 9.12. Furthermore, the system's more ordered appearance at lower temperatures can be noticed qualitatively.

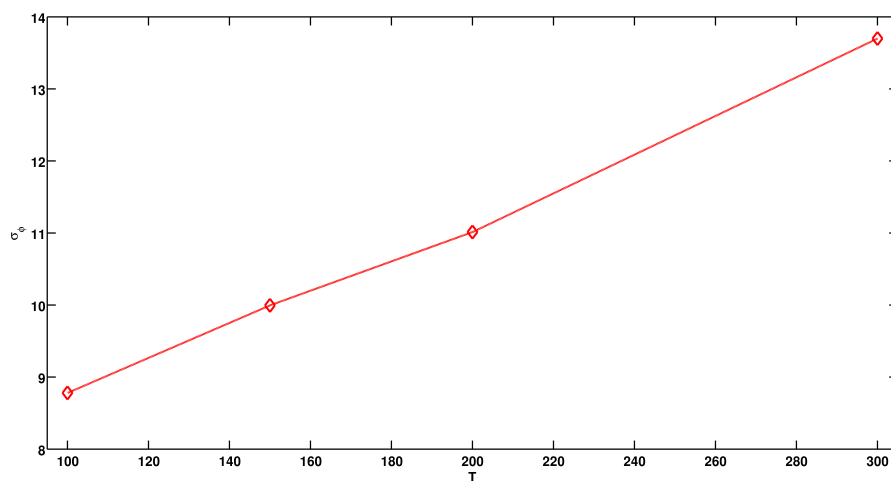
9.1. Biphenyldithiol SAM on a Au(111) surface



(a)



(b)



(c)

Figure 9.13.: Standard deviation of the distribution of the (a) tilt, (b) twist and (c) azimuthal angle of biphenyldithiol plotted versus temperature

9. Full coverage systems

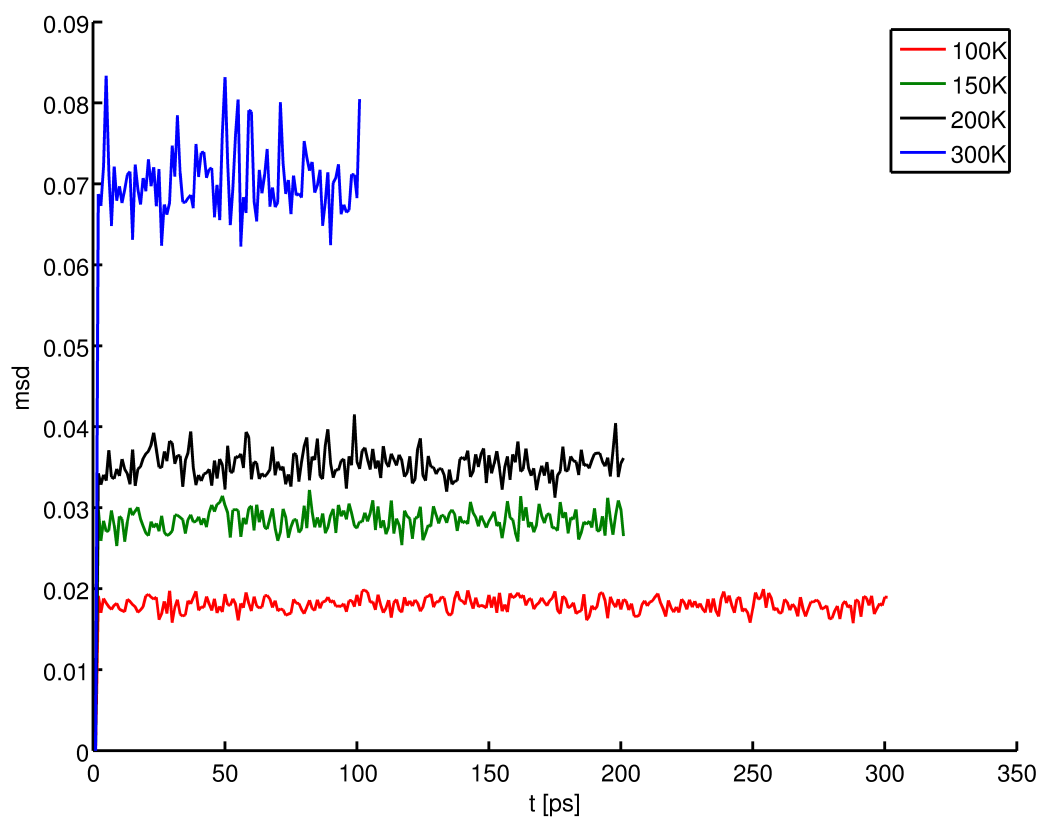


Figure 9.14.: Comparison of the MSD of the docking atoms at different temperatures of a biphenyldithiol SAM

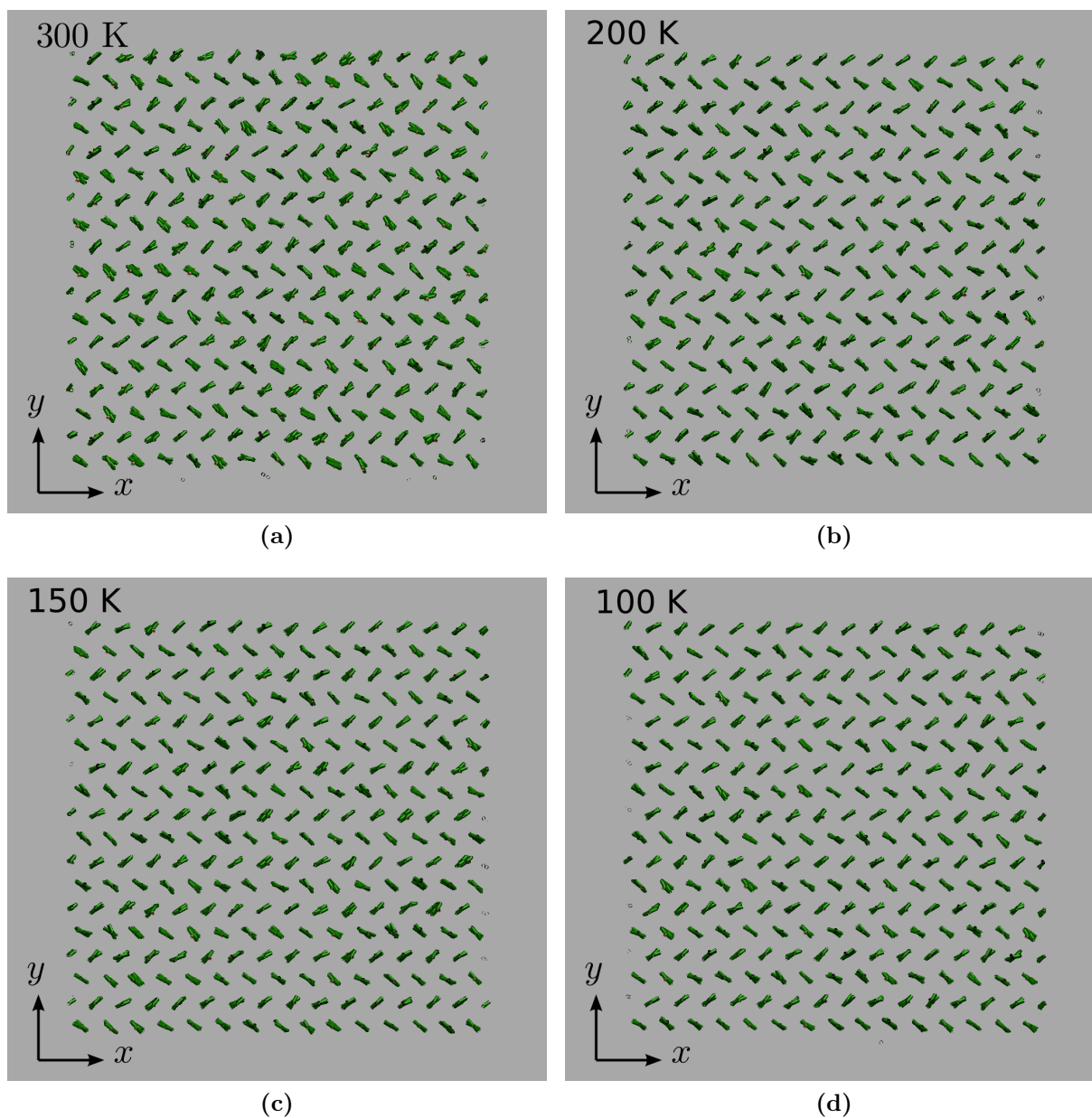


Figure 9.15.: Snapshots of the simulation cell during NVE equilibration of biphenyldithiol on gold at different temperatures; only the carbon rings and sulfur atoms of the SAM molecules are drawn.

Table 9.5.: *Expectation values and standard deviations of the angle distributions of Biphenylthiol at 300 K, 200 K, 150 K and 100 K*

	Θ	σ_{Θ}	Ψ_1	σ_{Ψ_1}	Ψ_2	σ_{Ψ_2}	ϕ	σ_{ϕ}
	
	tilt angle		twist angle		azimuthal angle		standard deviation	
$T[K]$	Θ	σ_{Θ}	Ψ_1	σ_{Ψ_1}	Ψ_2	σ_{Ψ_2}	ϕ	σ_{ϕ}
300	13.5	2.5	51.3	8.7	128.0	8.5	179.7	12.0
200	14.5	2.1	51.7	7.4	127.8	7.5	179.8	9.5
150	15.0	1.9	51.8	6.9	127.7	6.9	179.7	8.0
100	15.5	1.6	51.9	6.4	127.5	6.4	179.6	6.6

9.2. Biphenylthiol SAM on a Au(111) surface

The system investigated next was biphenylthiol on a gold (111) surface. A single biphenylthiol molecule with the assigned AM1-BCC charges is depicted in Figure 6.2a. The simulation setup was analog to system discussed in the previous section, i.e. the simulation times are equal to the values in Table 9.3.

The behavior of the fully covered system with respect to different temperatures turned out to be quite alike to the situation of the biphenyldithiol-gold system. For the sake of completeness the results are briefly discussed. The angle distributions get sharper at lower temperatures and in the case of tilt angle there is also a shift of the expectation values at different temperatures as in the biphenyldithiol system. In Figure 9.16 the angle distributions are shown. It can be seen that there is a narrowing of the peaks at lower temperatures being illustrated more clearly in Figure 9.17. The expectation values and standard deviations of the distributions in Figure 9.16 are listed in Table 9.5.

The MSD, depicted in Figure 9.18, shows similar features as in the case of biphenyldithiol. It also drops to lower values with decreasing temperatures. At lower temperatures there are also smaller oscillations like in the case of biphenyldithiol.

The docking atoms in the system of biphenylthiol are mostly located on *fcc-hollow* sites as in the case of biphenyldithiol, illustrated in Figure 9.19. Figure 9.20 shows a histogram of the distances traveled by the docking atoms. It can be found, that the majority of the docking groups hardly moves but stay around their initial docking sites.

Eventually, snapshots of the simulation cell at the observed temperatures are illus-

9.2. Biphenylthiol SAM on a Au(111) surface

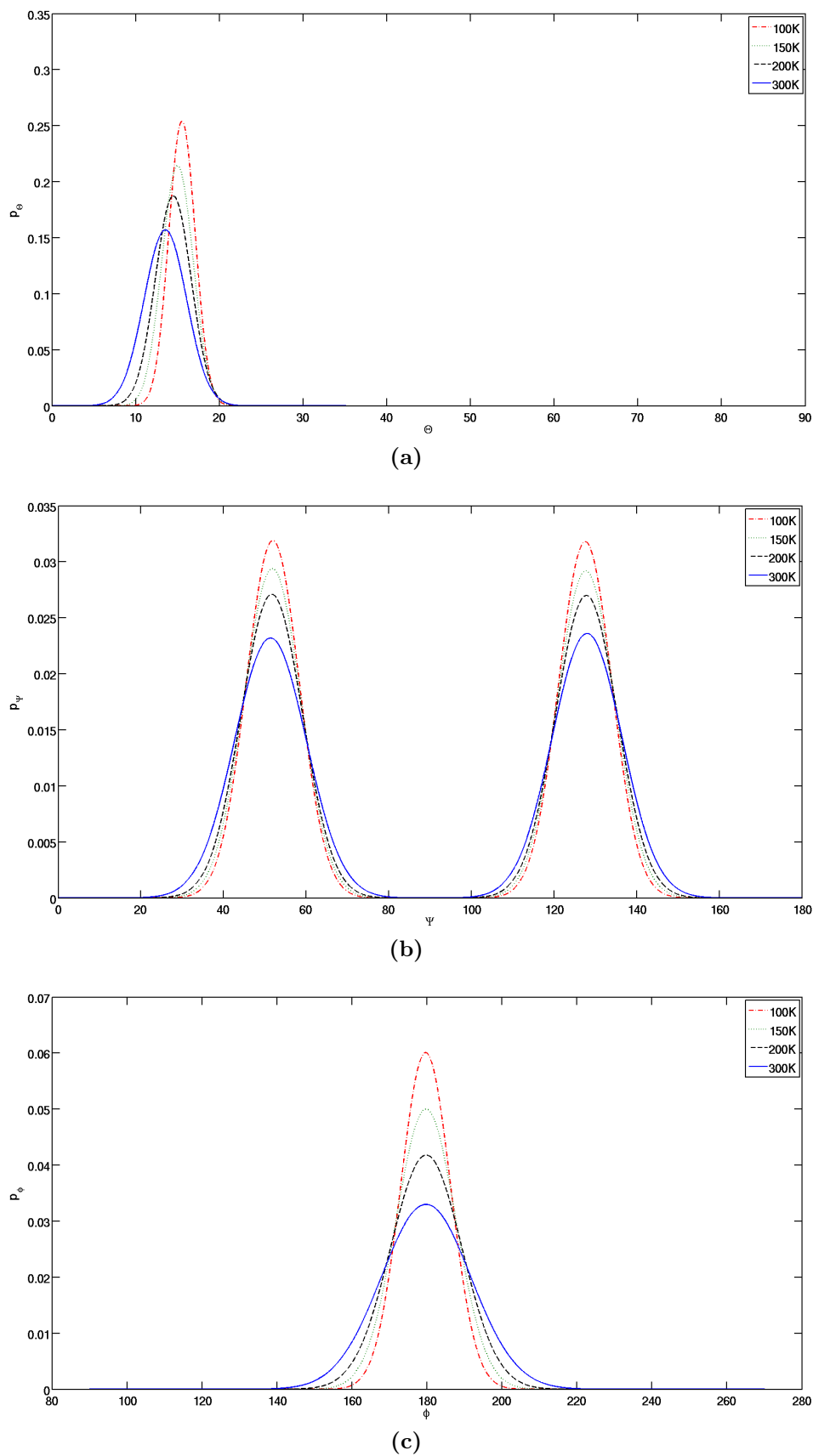
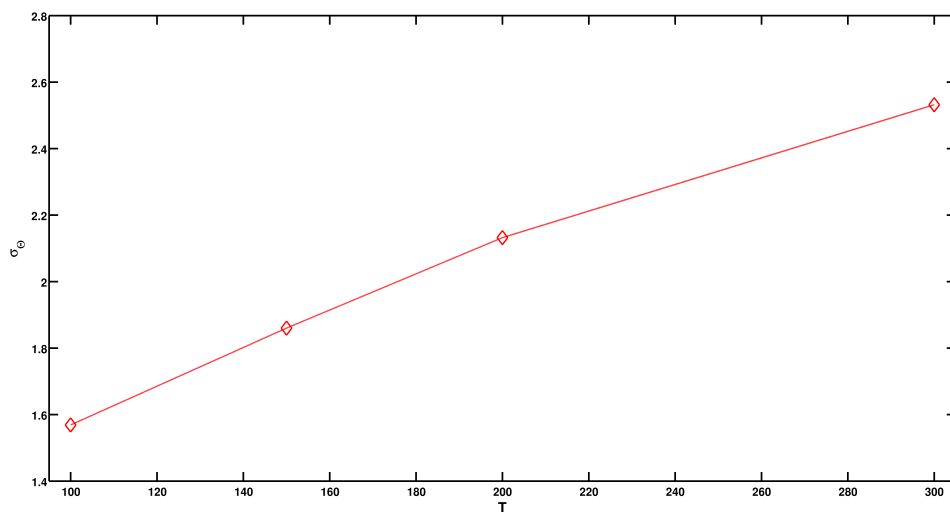
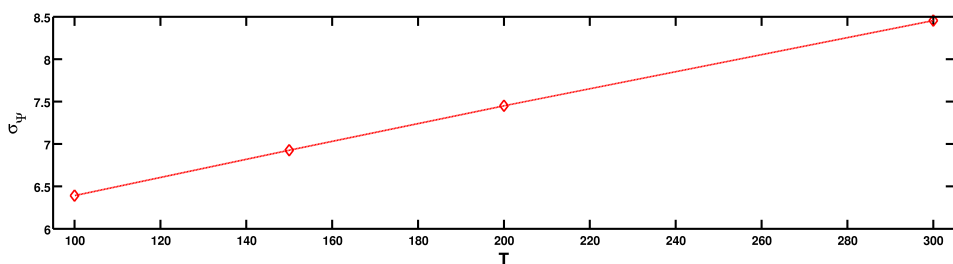
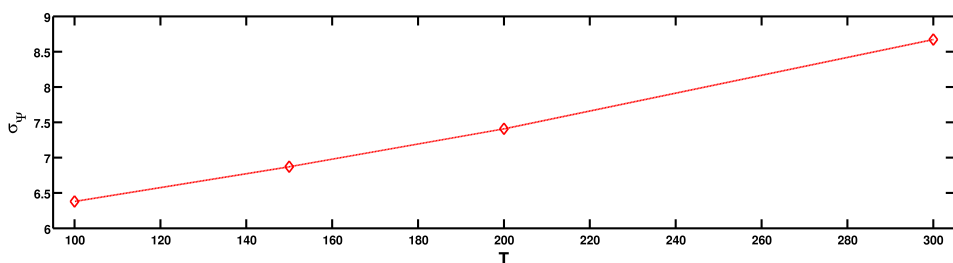


Figure 9.16.: Distributions of the (a) tilt, (b) twist and (c) azimuthal angle of biphenylthiol for different temperatures

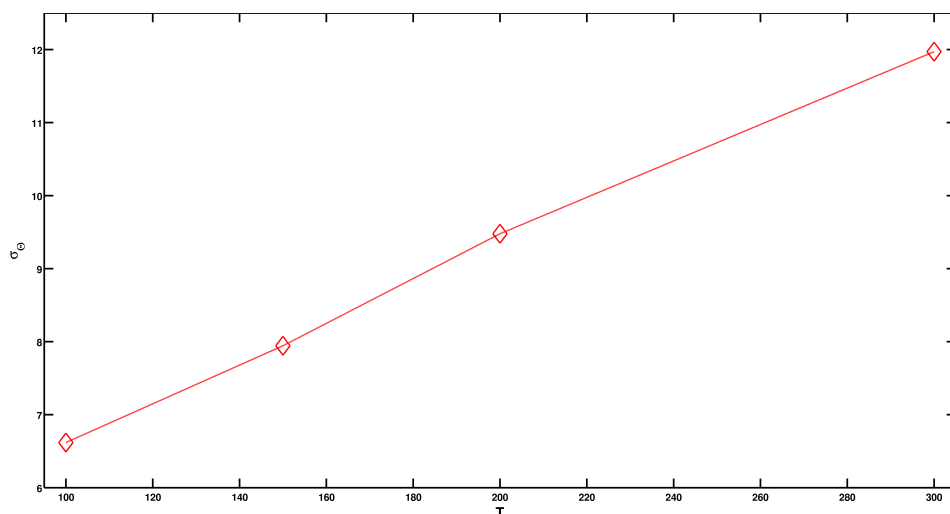
9. Full coverage systems



(a)



(b)



(c)

Figure 9.17.: Standard deviation of the distribution of the (a) tilt, (b) twist and (c) azimuthal angle of biphenylthiol plotted versus temperature

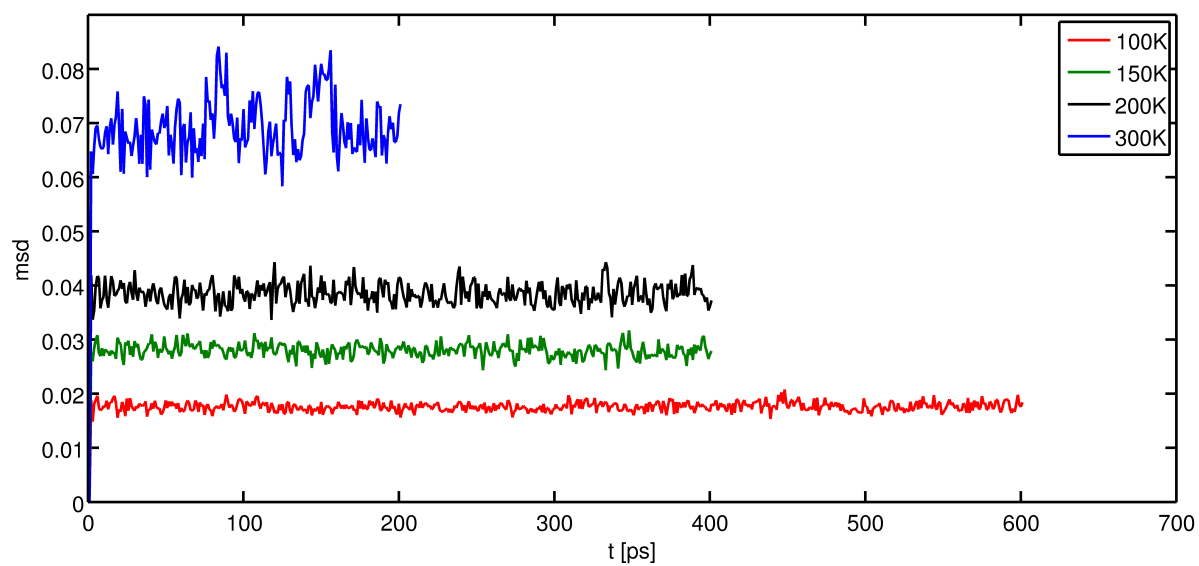


Figure 9.18.: Comparison of the MSD of the docking atoms of a biphenylthiol SAM at different temperatures

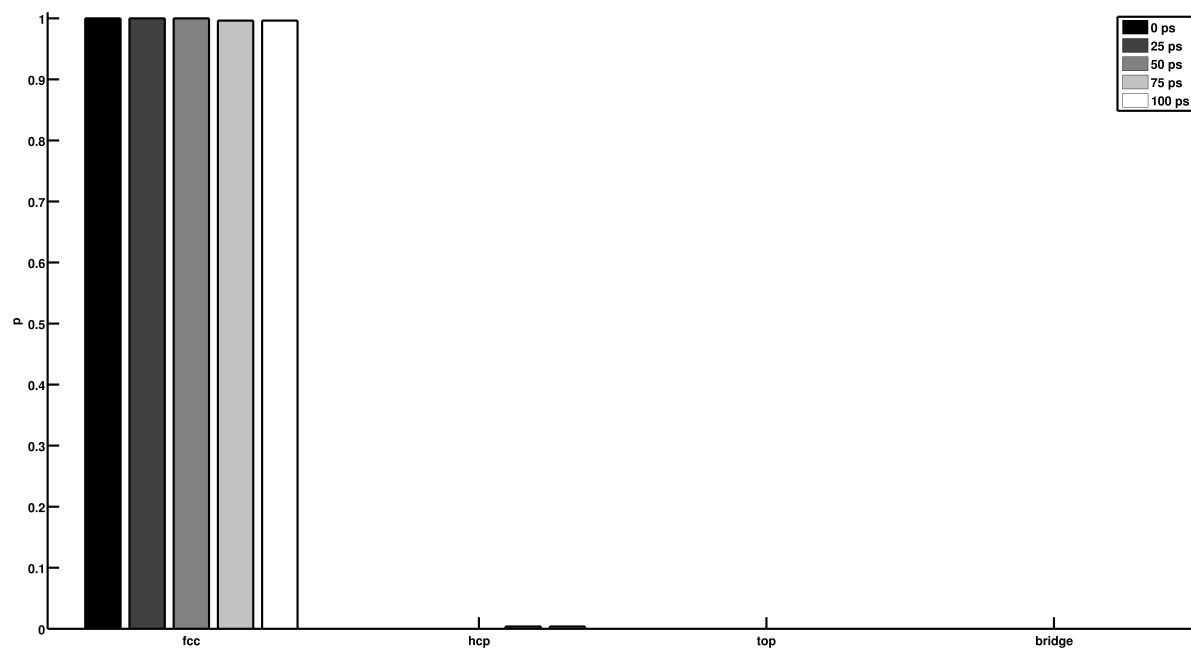


Figure 9.19.: Occupied docking sites of the biphenylthiol monolayer during NVE equilibration at 300 K

9. Full coverage systems

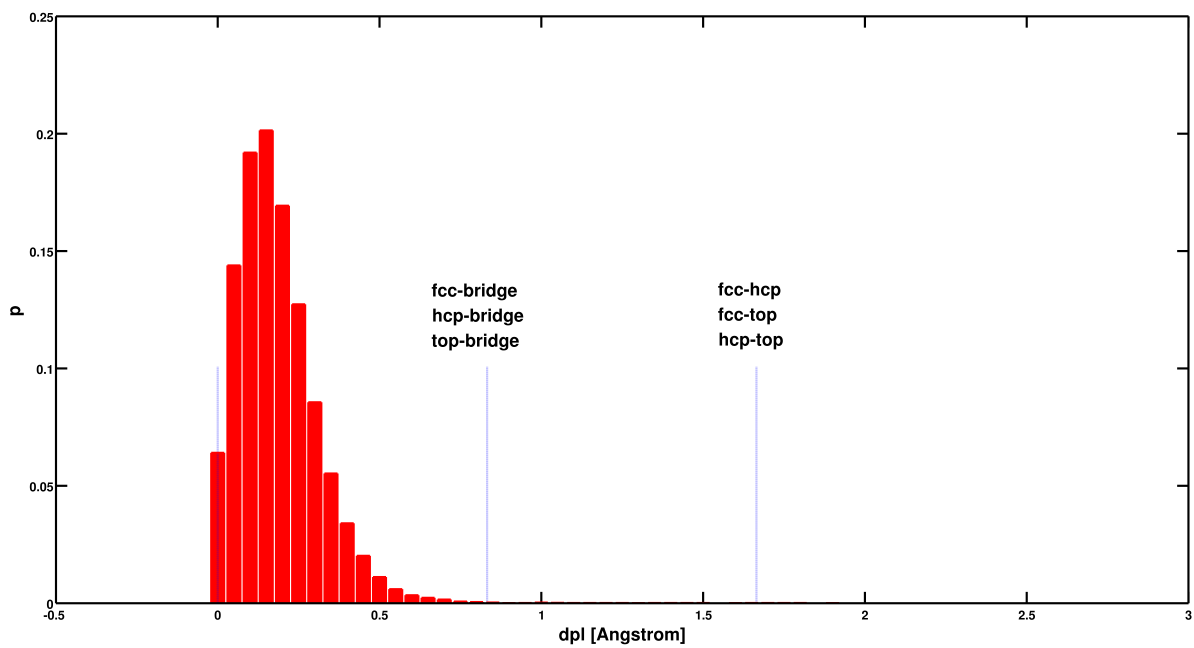


Figure 9.20.: Histogram of the displacements of the docking atoms during NVE equilibration at 300 K for biphenylthiol

trated in Figure 9.21. Like in the biphenyldithiol system a herringbone pattern emerges and a more ordered appearance is noticeable at lower temperatures.

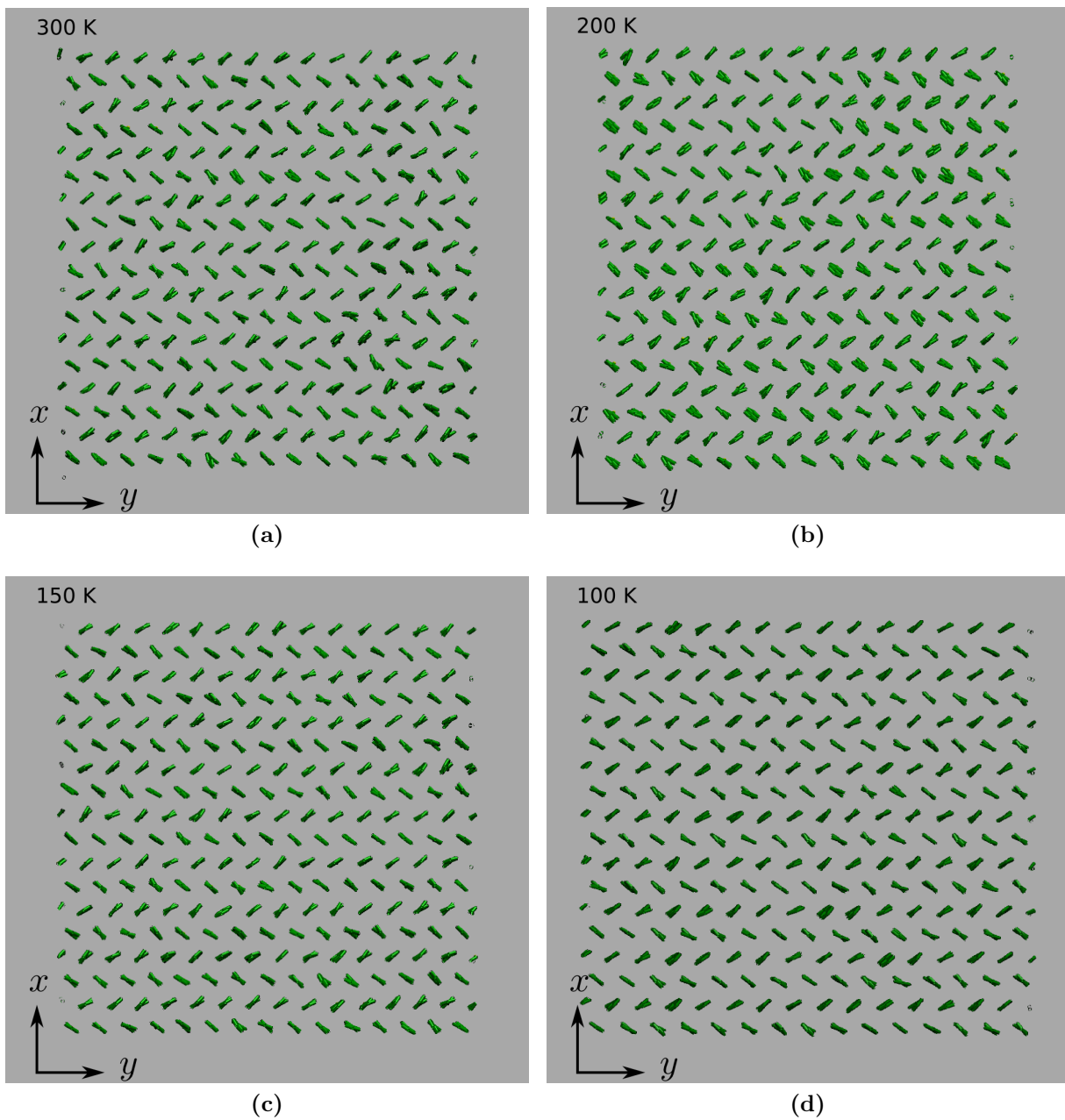


Figure 9.21.: Snapshots of the simulation cell of biphenylthiol on gold during NVE equilibration at different temperatures; only the carbon rings and sulfur atoms of the SAM molecules are drawn.

10. Low coverage systems

Until this point only systems with perfectly covered surfaces have been considered. As a next step SAMs, of biphenyldithiol and biphenylthiol with vacancies were investigated. In order to do so, the docking sites to be vacant were chosen randomly in such fashion that the resulting coverage was 90%. A routine to generate vacant docking sites was implemented in the *build layer* and *replicate* script, respectively (see Appendix 3). The simulations were analogous to those described in the previous chapters but only investigations at 300 K were conducted. The data for the displacement analysis was taken during equilibration in the NVE ensemble for 500 ps.

In Figure 10.1, the tilt angle distributions of biphenylthiol and biphenyldithiol SAMs are shown. One significant difference is that in the case of biphenyldithiol there are molecules lying horizontally on the gold surface associated with the small peak at around 90 degrees in the angle distribution. Since the thiol head group of biphenyldithiol represents another docking group the tendency of these molecules to lie down is comprehensible due to strong gold sulfur interactions. In the case of biphenylthiol there is no thiol head group but just a hydrogen atom hardly interacting with gold leading to quite tilted but still standing molecules.

Compared to the systems with fully covered surfaces (dashed dotted curves in Figure 10.1) the distributions are no longer Gaussian-like. The deviation from the Gaussian shape indicate that the molecules tend to tilt more than in fully covered systems. In the case of biphenyldithiol it seems that, mostly molecules in the vicinity of the lying molecules are more tilted. This leads to more or less distinguished peak and a shoulder at higher angles in the distribution. In the biphenylthiol system, the distribution is broader and asymmetric with respect to higher angles. This suggests that throughout the monolayer, the molecules are more tilted compared to the full coverage system. In Figure 10.7 the characteristics described above are illustrated.

The twist angle distributions for lowly covered biphenylthiol and biphenyldithiol sur-

10. Low coverage systems

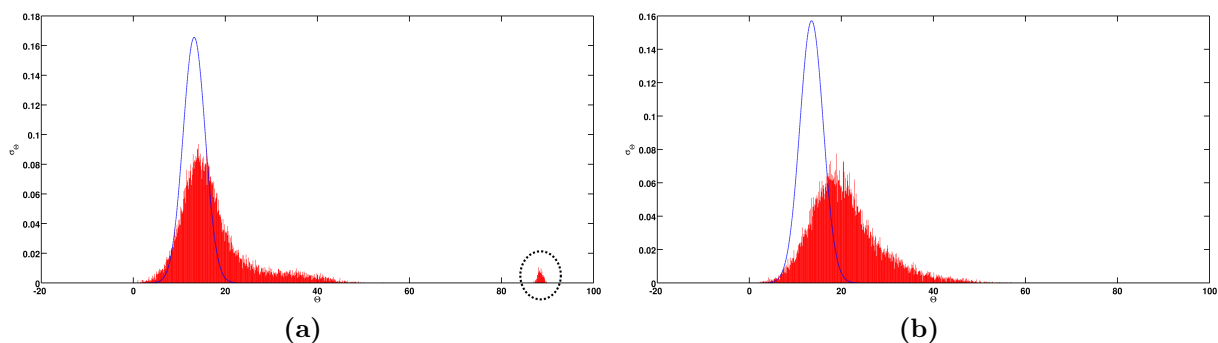


Figure 10.1.: Distributions of the tilt angle of a bpihenyldithiol monolayer (a) and a biphenylthiol monolayer (b) for 90% coverage of the surface. The dashed lines represent the distributions of fully covered systems. In picture (a) the peak representing the lying molecules is encircled.

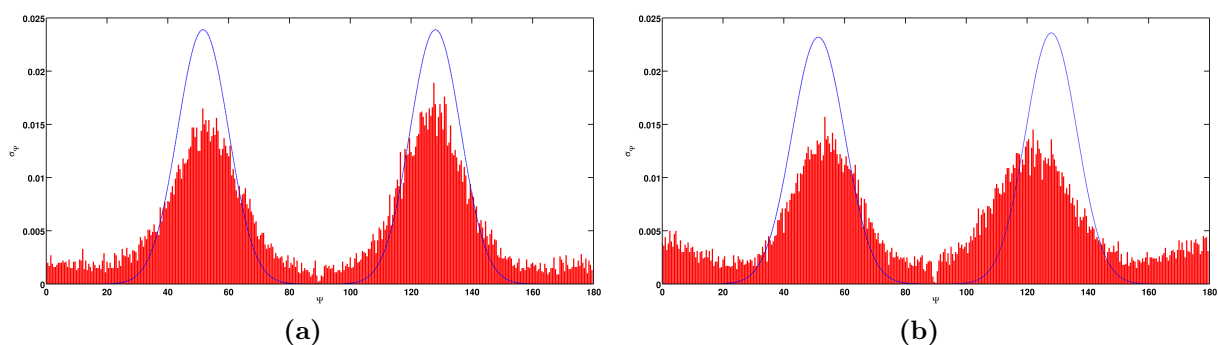


Figure 10.2.: Distributions of the twist angles of a bpihenyldithiol monolayer (a) and a biphenylthiol monolayer (b) for 90% coverage of the surface. The dashed lines represent the distributions of fully covered systems.

faces, as depicted in figure 10.2, show quite the same patterns. In comparison to full coverage systems, indicated by dotted lines in Figure 10.2, there are still two distinguished peaks but not that sharp anymore. In addition, there is a peak emerging in the biphenylthiol system at around 0° and 180° respectively. This third peak is associated with molecules whose phenyl rings are parallel to the $y - z$ plane.

The azimuthal angle in the case of biphenylthiol behaves similar in low coverage and full coverage systems (see Figure 10.3b). The distribution gets broader at lower coverages and the position of the distribution's peak is shifted about 15° with respect to the fully covered monolayer.

In the bpihenyldithiol system the situation is different, as depicted in Figure 10.3a,

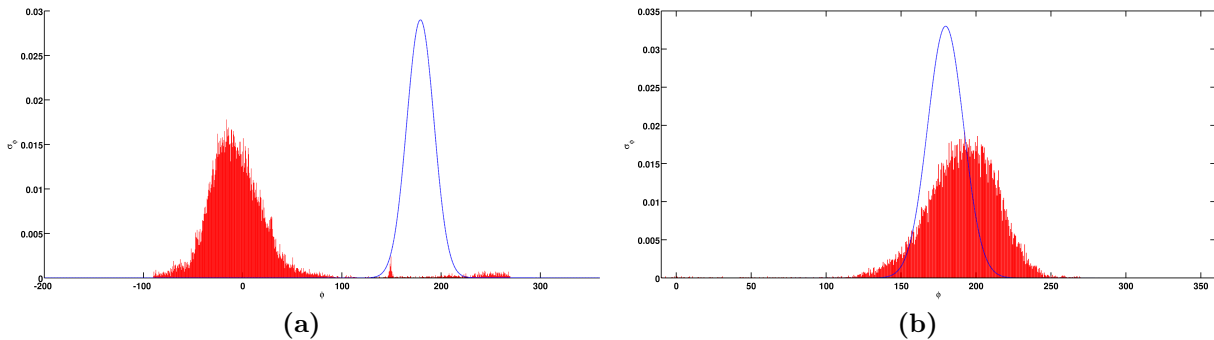


Figure 10.3.: Distributions of the azimuthal angles of a biphenyldithiol monolayer (a) and a biphenylthiol monolayer (b) for 90% coverage of the surface. The dashed lines represent the distributions of fully covered systems.

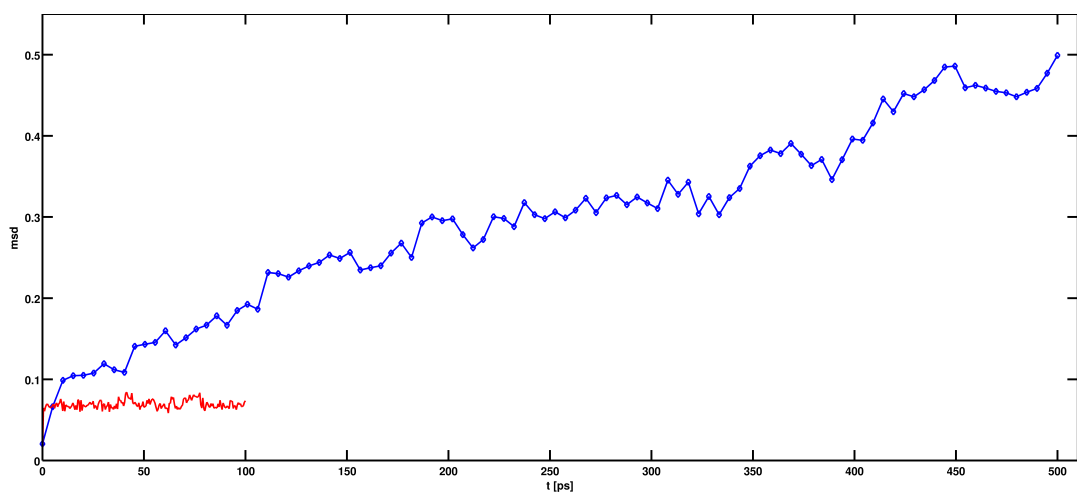
here the distribution is shifted about 180° , e.g. the molecules for the low coverage system tilt in the opposite direction than in the fully covered SAM. Additionally, there is small peak at about 150° and some entries at higher angles corresponding to lying molecules.

The MSD for low coverage SAMs is considerably higher than in fully covered systems and does not reach a constant level but grows continuously because the docking groups in the vicinity of the vacancies have more space to move, the MSD for biphenylthiol and biphenyldithiol is shown in Figure 10.4. A remarkable feature when comparing the biphenylthiol and biphenyldithiol system is the higher MSD of biphenylthiol within the considered timespan. This fact is understandable when considering the lying biphenyldithiol molecules on the lowly covered surface, these molecules block the rest of the standing molecules and restrain them from moving resulting in a lower MSD. The just discussed facts can be nicely seen in Figure 10.6 and 10.5, where the distributions of the docking atoms' displacement and the occupied docking sites are shown. While for biphenyldithiol at the beginning of the NVE equilibration most of the docking atoms were at *fcc* sites ($\sim 90\%$) and the rest at *hcp* sites, in the biphenylthiol system roughly half of the docking atoms were at *fcc* and *hcp* sites. This means that during the processes of cooling and equilibrating in the NVT ensemble 50 % of the docking groups changed its site with respect to the initial configuration (all molecules at *fcc* sites). Taking into account Figure 10.6 one can conclude that in the case of biphenylthiol the molecules mostly travel between *hcp* and *fcc* sites and that there is a small but recognizable transfer among *hcp* and *fcc* sites, respectively. In the biphenyldithiol system the only notable displacement happens between *fcc* and *hcp* sites.

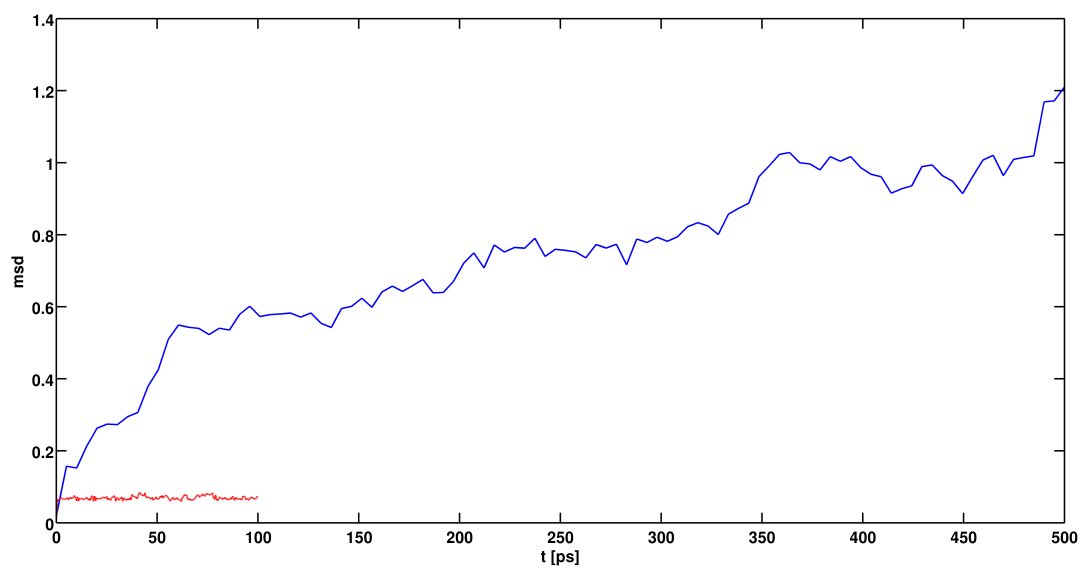
10. Low coverage systems

In the snapshots of the simulation cells (Figure 10.7) it can be found that in the case of biphenyldithiol the herringbone structure is conserved in certain domains being farther from the lying molecules, these domains are marked in Figure 10.7b. The areas of preserved herringbone structure lead to the more distinguished peaks in the twist angle's distribution (Figure 10.2) of biphenyldithiol compared to biphenylthiol.

In the depiction of the biphenylthiol layer (Figure 10.7a) molecules with a twist angle of roughly 90° , corresponding to the emerging third peak in the distribution in Figure 10.2a, can be observed. In contrast to biphenyldithiol, the lowly covered biphenylthiol SAM appears to be more disordered. While in the low coverage biphenyldithiol SAM disorder is mainly located in the proximate vicinity of the lying molecules, in the biphenylthiol system the shape of the herringbone structure is weaker over the whole layer. Thus the horizontal molecules act as a 'barrier' for disorder to propagate within a lowly covered SAM.



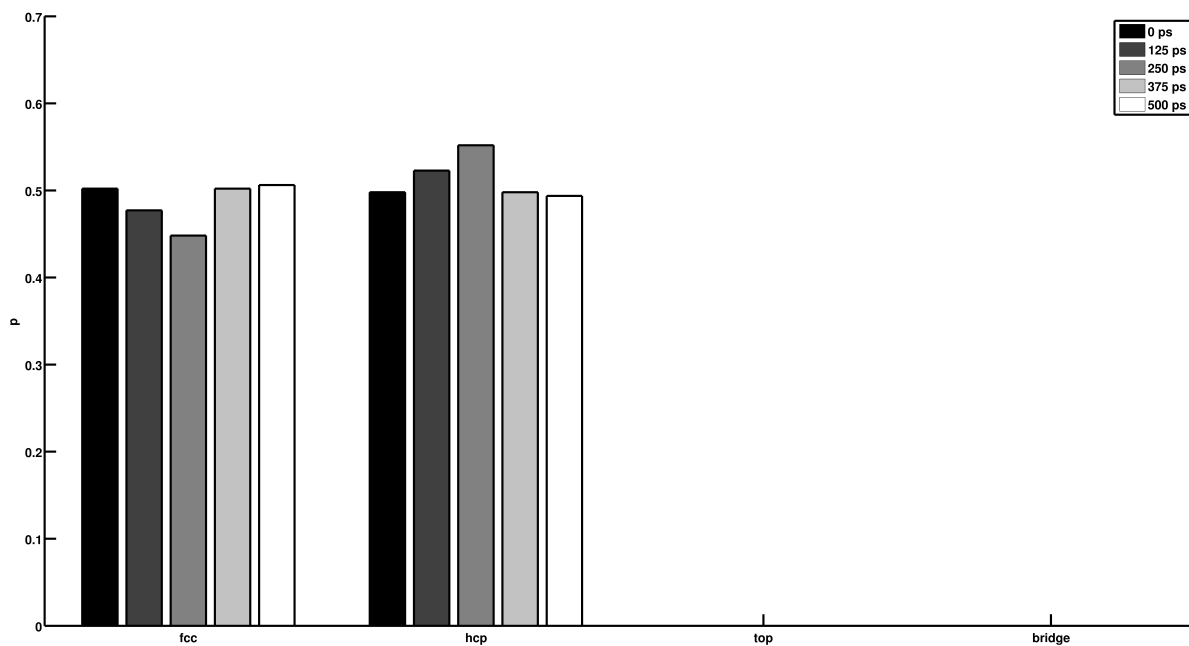
(a)



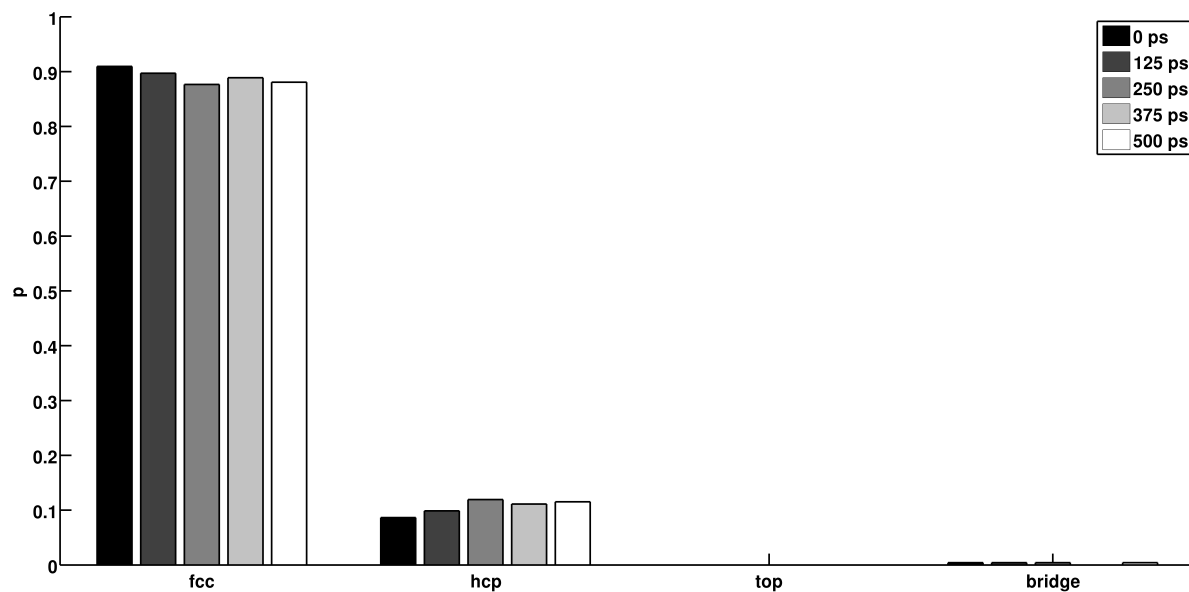
(b)

Figure 10.4.: Mean square displacements of the thiol docking groups in a 90 % covered monolayer of (a) biphenyldithiol and (b) biphenylthiol. The dashed lines indicate the MSD of the full coverage systems.

10. Low coverage systems



(a)



(b)

Figure 10.5.: Occupied docking sites of a (a) biphenylthiol and a (b) biphenyldithiol monolayer of 90 % coverage during NVE equilibration at 300 K.

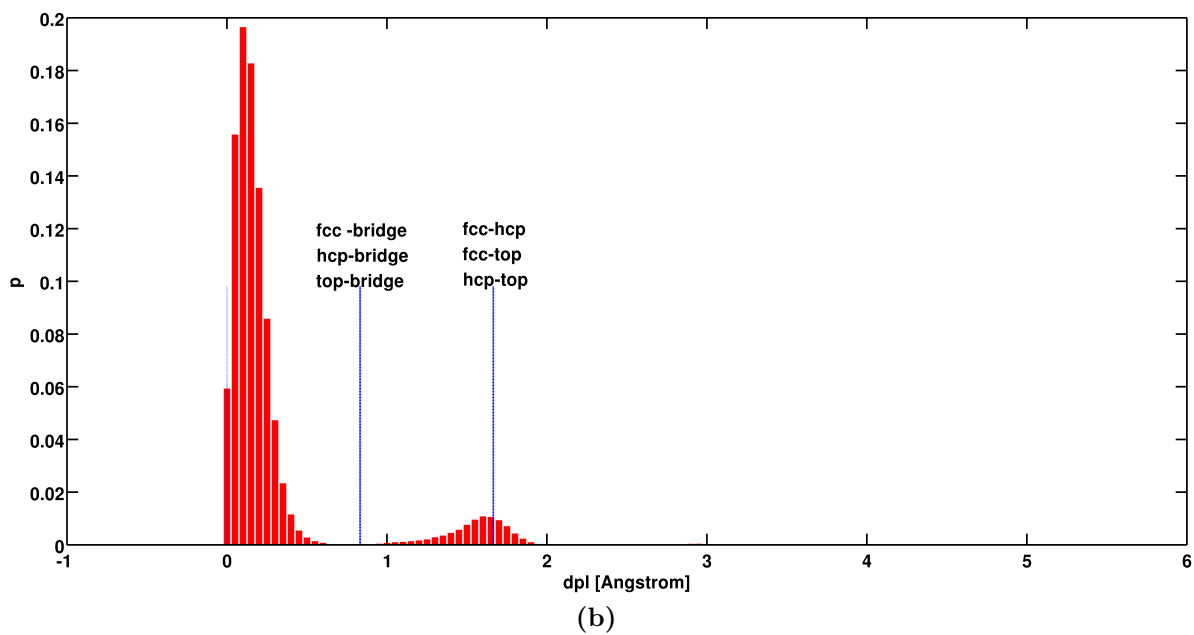
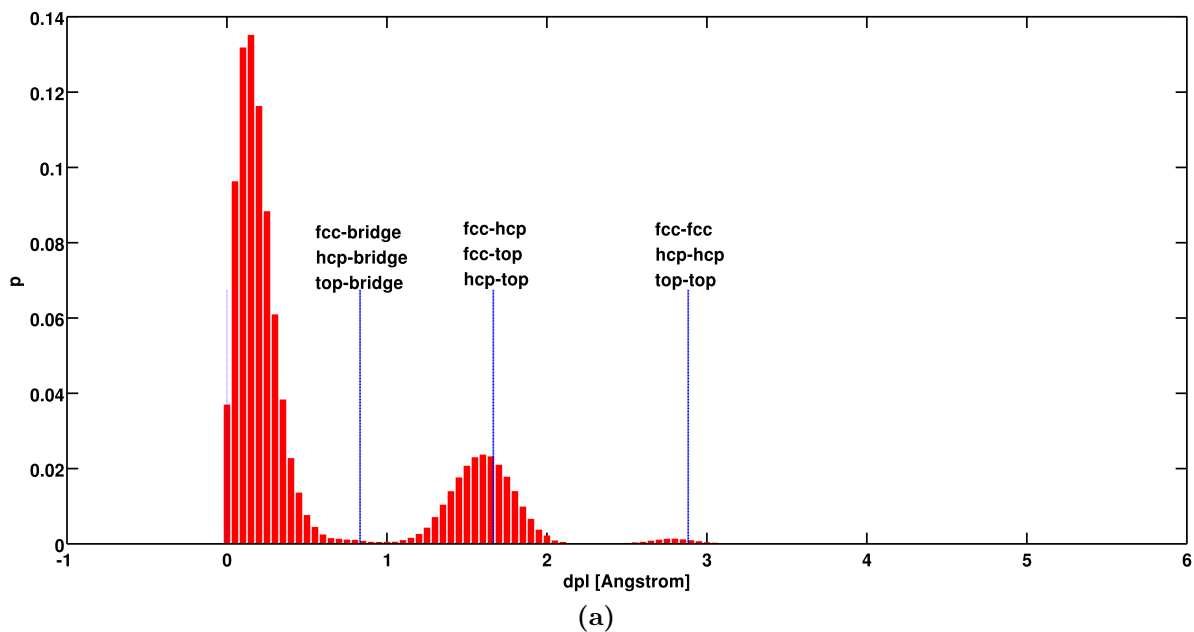


Figure 10.6.: *Distribution of the docking atoms' displacement during NVE equilibration at 300 K of a 90 % coverage SAM of (a) biphenylthiol and (b) biphenyldithiol. The vertical dotted lines indicate the distances between the described docking sites*

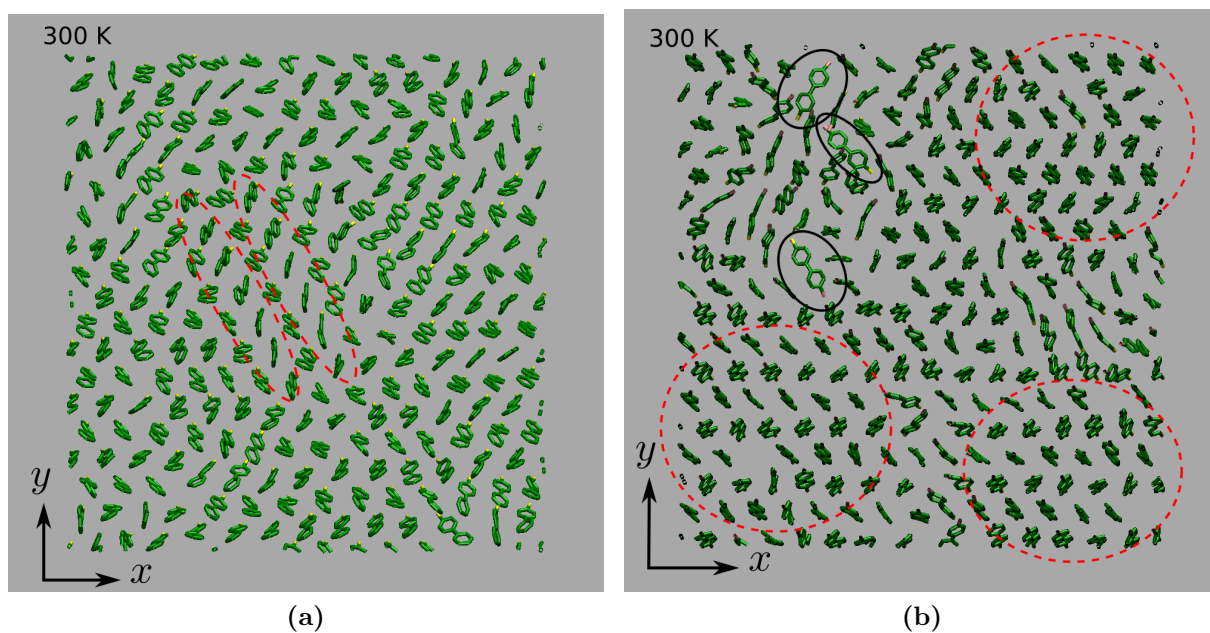


Figure 10.7.: Snapshots of the simulation cell of (a) biphenylthiol and (b) biphenyldithiol SAMs with 90 % coverage. The encircled regions in snapshot (a) refer to those molecules resulting in the third peak in the twist angle distribution (Figure 10.2). The lying biphenyldithiol molecules are marked by circles (solid line) in picture (b). The dashed circles in picture (b) mark the domains with preserved herringbone structure.

11. Negatively charged monolayers

As discussed in Chapter 6, there are basically two ways how to handle the charge of the hydrogen in the thiol docking group. In the calculations so far the positive atomic charge of the hydrogen has been added to the sulfur atom, resulting in an overall electrically neutral system. In this chapter calculations with negatively charged biphenyldithiol monolayers are presented and compared with simulations of neutrally charged SAMs.

Although this approach is physically doubtful, it leads to almost identical angle distributions for full coverage SAMs. This is illustrated in Figure 11.1, where the tilt, twist and azimuthal angle distributions for electrically neutral and negative full coverage SAMs are shown.

In the case of low coverage monolayers (90 %) the results for the negative and neutral monolayer differ as it can be seen in Figure 11.2.

Especially the twist angle distribution shows considerable differences among negative and neutral SAM. While in the neutral system the two peaks are just broadened with respect to the full coverage case, in the negative system a distinguished third peak emerges at 0° and 180° , respectively.

The azimuthal angle distribution is also significantly differing for the neutral and negative system. In the negative monolayer the distribution is considerably broadened compared to the neutral system. Furthermore, two peaks emerge in the angle distribution around 0° .

As mentioned above, leaving the molecules negatively charged is physically and computationally questionable. Thus no further investigations of such systems were conducted.

11. Negatively charged monolayers

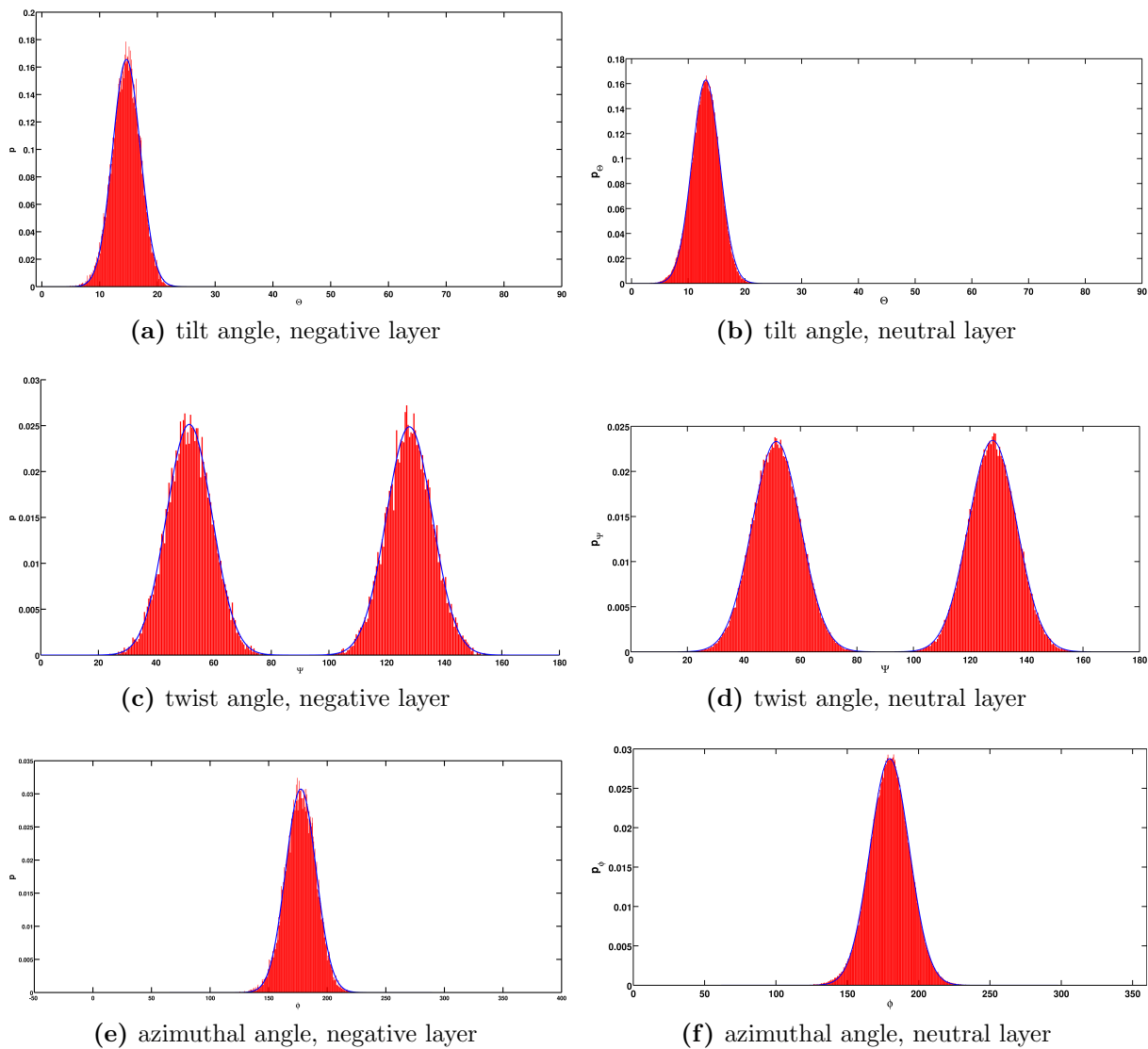
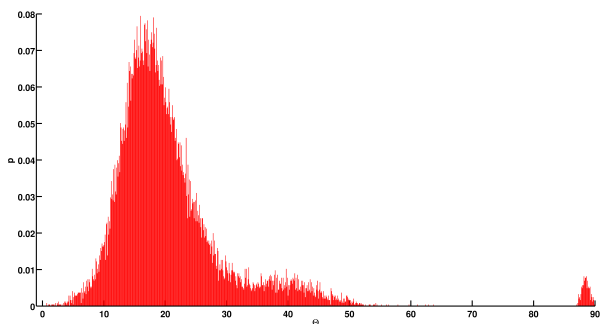
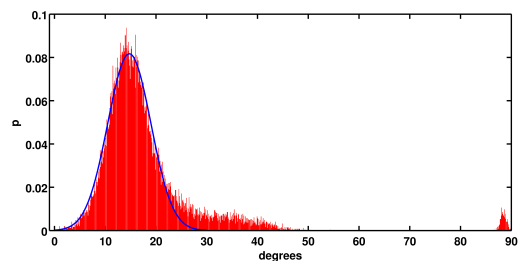


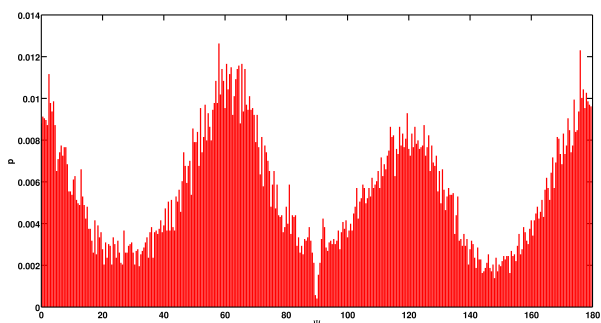
Figure 11.1.: Angle distributions of a neutral and negative biphenyldithiol SAM at full coverage.



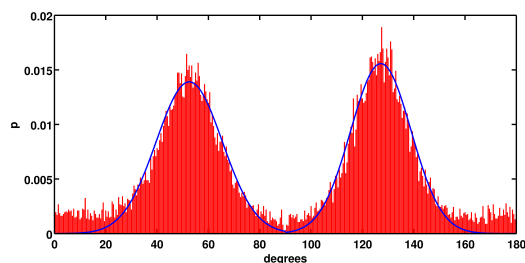
(a) tilt angle, negative layer



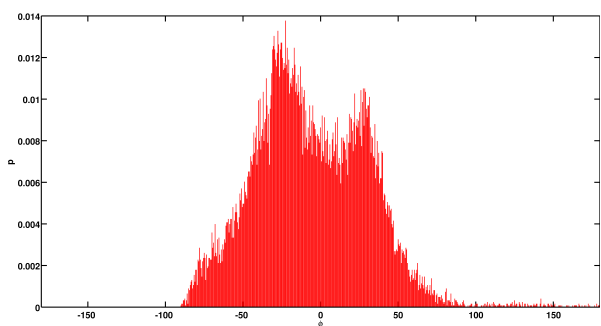
(b) tilt angle, neutral layer



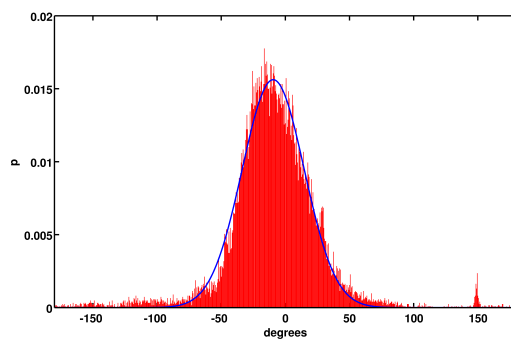
(c) twist angle, negative layer



(d) twist angle, neutral layer



(e) azimuthal angle, negative layer



(f) azimuthal angle, neutral layer

Figure 11.2.: Angle distributions of a neutral and negative biphenyldithiol SAM at 90 % coverage.

Bibliography

- [1] Godehard Sutmann. Classical molecular dynamics. *NIC Series*, 10:211 – 254, 2002. iii, 5, 9, 10, 23, 33
- [2] Taner E. Dirama and Joel A. Johnson. Conformation and dynamics of arylthiol self-assembled monolayers on au(111). *Langmuir*, 23(24):12208–12216, November 2007. ISSN 0743-7463. URL <http://dx.doi.org/10.1021/la7015963>. iii, 2, 32, 47, 55, 56, 61
- [3] Stefan K. Possanner, Karin Zojer, Peter Pacher, Egbert Zojer, and Ferdinand Schürerer. Threshold voltage shifts in organic thin-film transistors due to self-assembled monolayers at the dielectric surface. *Advanced Functional Materials*, 19(6):958–967, 2009. URL <http://dx.doi.org/10.1002/adfm.200801466>. 1
- [4] Andrew R. Leach. *Molecular modelling: principles and applications*. Peason Education Ltd., 2nd edition, 2001. 5, 9
- [5] L. Verlet. Computer experiments on classical fluids .i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159(1):98–&, 1967. 6
- [6] Junmei Wang, Romain M. Wolf, James W. Caldwell, Peter A. Kollman, and David A. Case. Development and testing of a general amber force field. *Journal of Computational Chemistry*, 25(9):1157–1174, 2004. URL <http://dx.doi.org/10.1002/jcc.20035>. 9, 14, 28, 31
- [7] J. E. Jones. On the determination of molecular fields. ii. from the equation of state of a gas. *Proceedings of the Royal Society of London. Series A*, 106(738):463–477, October 1924. URL <http://rspa.royalsocietypublishing.org/content/106/738/463.short>. 13
- [8] S. L. Mayo, B. D. Olafson, and W. A. Goddard. Dreiding - a generic force-field for molecular simulations. *Journal Of Physical Chemistry*, 94(26):8897–8909, December 1990. 13, 14, 31

Bibliography

- [9] R. A. Buckingham. The classical equation of state of gaseous helium, neon and argon. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 168(933):264–283, October 1938. URL <http://rspa.royalsocietypublishing.org/content/168/933/264.short>. 13
- [10] Micheal P. Allen. Introduction to molecular dynamics simulations. *NIC Series*, 23: 1 – 28, 2004. 15, 43
- [11] Chriatian Holm. Efficient methods for long range interactions in periodic geometries plus one application. *NIC Series*, 23:195 – 236, 2004. 15, 17
- [12] Jim Pollock, E.L. Glosli. Comments on p3m, fmm, and the ewald method fo large periodic coulomb systems. *Computer Physics Communications*, 95:93 – 110, 1996. 15
- [13] Peter Ewald. Die berechnung optischer und elektrostatischer gitterpotentiale. *Annalen der Physik*, 369:253–287, 1921. 16
- [14] J W Hockney, R W Eastwood. *Computer Simulations Using Particles*. Adam Hilger, Bristol and New York, 1988. 18
- [15] Markus Deserno and Christian Holm. How to mesh up ewald sums. ii. an accurate error estimate for the particle–particle–particle-mesh algorithm. *The Journal of Chemical Physics*, 109(18):7694–7701, 1998. URL <http://dx.doi.org/10.1063/1.477415>. 21, 41
- [16] Torsten Fließbach. *Statistische Physik*. Spektrum Akademischer Verlag GmbH, Heidelberg Berlin, 1999. 23
- [17] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 81(8):3684–3690, 1984. URL <http://dx.doi.org/10.1063/1.448118>. 24
- [18] Shuichi Nose. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 1984. URL <http://dx.doi.org/10.1063/1.447334>. 25
- [19] Shuichi Nosé. A molecular dynamics method for simulations in the canonical ensemble. *Molecular Physics: An International Journal at the Interface Between*

- Chemistry and Physics*, 52(2):255–268, 1984. URL <http://www.informaworld.com/10.1080/00268978400101201>. 25
- [20] William G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31(3):1695–, March 1985. URL <http://link.aps.org/abstract/PRA/v31/p1695>. 25
- [21] U. Chandra Singh and Peter A. Kollman. An approach to computing electrostatic charges for molecules. *Journal of Computational Chemistry*, 5(2):129–145, 1984. URL <http://dx.doi.org/10.1002/jcc.540050204>. 27
- [22] C. I. Bayly, P. Cieplak, W. D. Cornell, and P. A. Kollman. A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges - the resp model. *Journal Of Physical Chemistry*, 97(40):10269–10280, October 1993. 27
- [23] Araz Jakalian, Bruce L. Bush, David B. Jack, and Christopher I. Bayly. Fast, efficient generation of high-quality atomic charges. am1-bcc model: I. method. *Journal of Computational Chemistry*, 21(2):132–146, 2000. URL [http://dx.doi.org/10.1002/\(SICI\)1096-987X\(20000130\)21:2<132::AID-JCC5>3.0.CO;2-P](http://dx.doi.org/10.1002/(SICI)1096-987X(20000130)21:2<132::AID-JCC5>3.0.CO;2-P). 27
- [24] Araz Jakalian, David B. Jack, and Christopher I. Bayly. Fast, efficient generation of high-quality atomic charges. am1-bcc model: Ii. parameterization and validation. *Journal of Computational Chemistry*, 23(16):1623–1641, 2002. URL <http://dx.doi.org/10.1002/jcc.10128>. 27
- [25] Michael J. S. Dewar, Eve G. Zoebisch, Eamonn F. Healy, and James J. P. Stewart. Development and use of quantum mechanical molecular models. 76. am1: a new general purpose quantum mechanical molecular model. *Journal of the American Chemical Society*, 107(13):3902–3909, June 1985. ISSN 0002-7863. URL <http://dx.doi.org/10.1021/ja00299a024>. 27
- [26] Y. H. Kim, S. S. Jang, and W. A. Goddard. Conformations and charge transport characteristics of biphenyldithiol self-assembled-monolayer molecular electronic devices: A multiscale computational study. *Journal Of Chemical Physics*, 122(24):244703, June 2005. 31
- [27] S. S. Jang, Y. H. Jang, Y. H. Kim, W. A. Goddard, J. W. Choi, J. R. Heath, B. W. Laursen, A. H. Flood, J. F. Stoddart, K. Norgaard, and T. Bjornholm. Molecular

- dynamics simulation of amphiphilic bistable [2]rotaxane langmuir monolayers at the air/water interface. *Journal Of The American Chemical Society*, 127(42):14804–14816, October 2005. 31, 32
- [28] <http://lammps.sandia.gov/>. URL <http://lammps.sandia.gov/>. 32, 39, 43, 55
- [29] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1):1–19, March 1995. ISSN 0021-9991. URL <http://www.sciencedirect.com/science/article/B6WHY-45N1B-3N/2/58aa2a309d2ebbbe60e0f417d398b0ef>. 32
- [30] <http://ambermd.org/>. URL <http://ambermd.org/>. 35
- [31] Junmei Wang, Wei Wang, Peter A. Kollman, and David A. Case. Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics and Modelling*, 25(2):247–260, October 2006. ISSN 1093-3263. URL <http://www.sciencedirect.com/science/article/B6TGP-4J616VX-1/2/16a009bb4966316d9ffc60f355092909>. 35
- [32] Georg Heimel, Lorenz Romaner, Jean-Luc Bredas, and Egbert Zojer. Interface energetics and level alignment at covalent metal-molecule junctions: pi-conjugated thiols on gold. *Physical Review Letters*, 96(19):196806, 2006. URL <http://dx.doi.org/10.1103/PhysRevLett.96.196806>. 35
- [33] G. Heimel, L. Romaner, E. Zojer, and J. L. Bredas. Toward control of the metal-organic interfacial electronic structure in molecular electronics: A first-principles study on self-assembled monolayers of pi-conjugated molecules on noble metals. *Nano Letters*, 7(4):932–940, April 2007. 35
- [34] <http://www.cmbi.kun.nl/molden/molden.html>. URL <http://www.cmbi.kun.nl/molden/molden.html>. 44

Appendix

1. Input files

A typical input file for a simulation of biphenylthiol on Au(111) is shown below.

```
#thermalization of the Au - Bpt system

log    log.run_au_bpt_bcc_run1
units      real
boundary p p p
atom_style full
pair_style hybrid buck 14.0 lj/charmm/coul/long 14.0 16.0
bond_style harmonic
angle_style harmonic
dihedral_style harmonic
special_bonds amber
kspace_style pppm 1.0e-4

lattice fcc 4.0782 orient x -1 -1 2 orient y 1 -1 0 orient z 1 1 1 &
spacing 1 1 1
read_data data.layer_bpt_bcc_15x18_neutr

region au block INF INF INF INF -3.8 -0.5
create_atoms 5 region au

pair_coeff 1 5 buck 1.5e6 0.22 6723.8 # S-Au
pair_coeff 2 5 buck 10416.3 0.30 261.0 # C-Au
pair_coeff 3 5 buck 6673.0 0.26 70.3 # H-Au
pair_coeff 4 5 buck 6673.0 0.26 70.3 # H-Au
```

Appendix

```
neighbor 2.0 bin
neigh_modify every 1 delay 0 check yes exclude type 5 5

group metal type 5
group layer type 1 2 3 4
group dock type 1
group layer_noh type 1 2

#---- restraining the Au atoms from moving
velocity metal set 0 0 0
fix m metal setforce 0 0 0

#---- fixing the docking atoms
velocity dock set 0 0 0
fix fdock dock setforce 0 0 0

compute Temp_layer layer temp
compute E_kin_layer layer ke

thermo_style custom step temp epair emol etotal press c_Temp_layer &
c_E_kin_layer
thermo 500

#---- minimization of the initial configuration
minimize 1e-8 1e-8 1000 1000

#---- heat to 900 K
fix fshake layer shake 1e-4 20 0 t 3 4

fix 2 layer temp/berendsen 0 900 1
fix 3 layer nve
fix_modify 2 temp Temp_layer

timestep 1
```

```
run 200000

write_restart rst.au_bpt_bcc_run1_heat

unfix 2
unfix 3

#---- equilibrate at 900 K in NVT ensemble

fix 4 layer nvt 900 900 10

timestep 1
run 400000

write_restart rst.au_bpt_bcc_run1_eq_nvt_hot

unfix 4
unfix fdock

#---- cooling the system to 300 K
fix 5 layer temp/berendsen 900 300 1
fix 6 layer nve
fix_modify 5 temp Temp_layer

timestep 1
run 1500000

write_restart rst.au_bpt_bcc_run1_cool_300

unfix 5
unfix 6

# ---- Equilibration at 300 K in NVT ensemble
fix 7 layer nvt 300 300 10
fix s_coord dock coord/original
```

Appendix

```
fix msd_dock dock msd 1000 msd.dock_au_bpt_bcc_eq_nvt_300
```

```
compute sdpl dock displace/atom s_coord
```

```
dump s dock custom 1000 sdpl.au_bpt_bcc_eq_nvt_300 &  
c_sdpl[1] c_sdpl[2] c_sdpl[3] c_sdpl[4]
```

```
dump catom layer_noh custom 1000 &  
layer.au_bpt_bcc_eq_nvt_300 id mol type x y z
```

```
timestep 1
```

```
run 100000
```

```
write_restart rst.au_bpt_bcc_run1_nvt_300
```

```
unfix 7
```

```
unfix s_coord
```

```
unfix msd_dock
```

```
uncompute sdpl
```

```
undump s
```

```
undump catom
```

```
#---- Equilibration in NVE ensemble
```

```
fix 8 layer nve
```

```
fix s_coord dock coord/original
```

```
fix msd_dock dock msd 500 msd.dock_au_bpt_bcc_eq_nve_300
```

```
compute sdpl dock displace/atom s_coord
```

```
dump s dock custom 500 sdpl.au_bpt_bcc_eq_nve_300 &  
c_sdpl[1] c_sdpl[2] c_sdpl[3] c_sdpl[4]
```

```
dump catom layer_noh custom 1000 &  
layer.au_bpt_bcc_eq_nve_300 id mol type x y z
```

```
timestep 1
run 100000

write_restart rst.au_bpt_bcc_run1_nve_300

unfix 8
unfix s_coord
unfix msd_dock
uncompute sdpl
undump s
undump catom
```

2. Data file

In the following a commented data file is presented. For the sake of clarity the sections 'Atoms', 'Bonds', 'Angles' and 'Dihedrals' were truncated after first molecule's data, respectively.

```
LAMMPS data file generated from data.bpt_bcc_neutr
# First line is ignored

5940 atoms # number of atoms
6210 bonds # number of bonds
9720 angles # number of angles
17280 dihedrals #number of dihedrals
0 impropers

5 atom types # number of atom types
5 bond types # number of bond types
```

Appendix

```
7 angle types # number of angle types
3 dihedral types # number of dihedral types

0 74.9213 xlo xhi # dimensions of the central cell in x-direction
0 77.8605 ylo yhi # dimensions of the central cell in y-direction
-50 150 zlo zhi # dimensions of the central cell in z-direction
```

Masses

```
1 32.060000000000002 # mass of atom type 1 (S)
2 12.01 # mass of atom type 2 (C)
3 1.008 # mass of atom type 3 (H)
4 1.008 # mass of atom type 4 (H)
5 196.96655 # mass of atom type 5 (Au)
```

Pair Coeffs

```
#pair coeff. for the vdW interactions involving atom type 1:
1 lj/charmm/coul/long 0.25 3.5635948725613571
```

```
#pair coeff. for the vdW interactions involving atom type 2:
2 lj/charmm/coul/long 0.086000000128358844 3.3996695079448309
```

```
#pair coeff. for the vdW interactions involving atom type 3:
3 lj/charmm/coul/long 0.015000000064220668 2.5996424587350853
```

```
#pair coeff. for the vdW interactions involving atom type 4:
4 lj/charmm/coul/long 0.015000000064220668 2.5996424587350853
```

```
#pair coeff. for the vdW interactions involving atom type 5
#(set to some arbitrary number because turned off or defined
#specifically in the input file):
5 lj/charmm/coul/long 0 0
```

Bond Coeffs

```
1 245.80000000000001 1.7869999999999999 # bond coeff. for bond type 1
2 478.39999999999998 1.387 # bond coeff. for bond type 2
3 344.30000000000001 1.087 # bond coeff. for bond type 3
4 466.10000000000002 1.395 # bond coeff. for bond type 4
5 346.5 1.49 # bond coeff. for bond type 5
```

Angle Coeffs

```
1 61.5 120.13005154209219 # angle-bending coeff. for type 1
2 67.200000000000003 119.97005135892852 # angle-bending coeff. for type 2
3 48.5 120.01005126147997 # angle-bending coeff. for type 3
4 67.200000000000003 119.07005097321041 # angle-bending coeff. for type 4
5 67.099999999999994 118.75005117984087 # angle-bending coeff. for type 5
6 62.600000000000001 127.01005426150974 # angle-bending coeff. for type 6
7 48.0 121.0800516627602 # angle-bending coeff. for type 7
```

Dihedral Coeffs

```
1 3.625 -1 2.0 # dihedral coeff. for type 1
2 1.0 -1 2.0 # dihedral coeff. for type 2
3 1.1000000000000001 -1 2.0 # dihedral coeff. for type 3
```

Atoms

```
# atom number | molecule number | atom type | charge | x | y | z
1 0 1 -0.0589 0 0 0
2 0 2 0.0165 0.00542896 0.00255468 1.71
3 0 2 -0.1192 -1.08941 -0.512638 2.414
4 0 2 -0.1028 -1.08489 -0.510509 3.814
5 0 2 -0.0439 0.0135724 0.00638669 4.51
6 0 2 -0.101 1.10841 0.52158 3.806
7 0 2 -0.1459 1.10479 0.519877 2.406
```


Appendix

8 0 3 0.1445 -1.94447 -0.915 1.873
9 0 3 0.1392 -1.93723 -0.911593 4.362
10 0 2 -0.0379 0.0190014 0.00894137 5.96
11 0 3 0.1381 1.96438 0.924367 4.347
12 0 3 0.1306 1.95624 0.920535 1.858
13 0 2 -0.1191 1.11746 0.525837 6.656
14 0 2 -0.1301 1.12199 0.527966 8.056
15 0 2 -0.1269 0.0271448 0.0127734 8.76
16 0 2 -0.1301 -1.07222 -0.504548 8.064
17 0 2 -0.1186 -1.07674 -0.506677 6.664
18 0 3 0.1341 1.9689 0.926496 6.108
19 0 3 0.1323 1.97705 0.930328 8.597
20 0 3 0.1318 0.0307641 0.0144765 9.849
21 0 3 0.1324 -1.92366 -0.905207 8.612
22 0 4 0.1349 -1.93181 -0.909039 6.123
.
.
.

Bonds

#bond number | bond type | atom numbers of involved atoms
1 3 3 8
2 3 4 9
3 3 6 11
4 3 7 12
5 3 13 18
6 3 14 19
7 3 15 20
8 3 16 21
9 3 17 22
10 1 1 2
11 2 2 3
12 2 2 7
13 2 3 4

14 4 4 5
15 4 5 6
16 5 5 10
17 2 6 7
18 4 10 13
19 4 10 17
20 2 13 14
21 2 14 15
22 2 15 16
23 2 16 17
.
.
.

Angles

#angle number | angle type | atom numbers of involved atoms
1 3 2 3 8
2 3 2 7 12
3 3 3 4 9
4 3 4 3 8
5 7 5 4 9
6 7 5 6 11
7 3 6 7 12
8 3 7 6 11
9 7 10 13 18
10 7 10 17 22
11 3 13 14 19
12 3 14 13 18
13 3 14 15 20
14 3 15 14 19
15 3 15 16 21
16 3 16 15 20

Appendix

17 3 16 17 22
18 3 17 16 21
19 1 1 2 3
20 1 1 2 7
21 2 2 3 4
22 2 2 7 6
23 2 3 2 7
24 4 3 4 5
25 5 4 5 6
26 6 4 5 10
27 4 5 6 7
28 6 5 10 13
29 6 5 10 17
30 6 6 5 10
31 4 10 13 14
32 4 10 17 16
33 5 13 10 17
34 2 13 14 15
35 2 14 15 16
36 2 15 16 17
.
.
.

Dihedrals

#dihedral number		dihedral type		atom numbers of involved atoms
1		1		1 2 3 8
2		1		1 2 7 12
3		1		2 3 4 9
4		1		2 7 6 11
5		1		3 2 7 12
6		1		4 5 6 11
7		1		5 4 3 8
8		1		5 6 7 12

9 1 5 10 13 18
10 1 5 10 17 22
11 1 6 5 4 9
12 1 7 2 3 8
13 1 8 3 4 9
14 1 9 4 5 10
15 1 10 5 6 11
16 1 10 13 14 19
17 1 10 17 16 21
18 1 11 6 7 12
19 1 13 10 17 22
20 1 13 14 15 20
21 1 14 15 16 21
22 1 15 14 13 18
23 1 15 16 17 22
24 1 16 15 14 19
25 1 17 10 13 18
26 1 17 16 15 20
27 1 18 13 14 19
28 1 19 14 15 20
29 1 20 15 16 21
30 1 21 16 17 22
31 3 2 4 3 8
32 3 3 5 4 9
33 3 7 5 6 11
34 3 2 6 7 12
35 3 14 10 13 18
36 3 13 15 14 19
37 3 14 16 15 20
38 3 15 17 16 21
39 3 16 10 17 22
40 1 1 2 3 4
41 1 1 2 7 6
42 1 2 3 4 5
43 1 2 7 6 5

Appendix

44 1 3 2 7 6
45 1 3 4 5 6
46 1 3 4 5 10
47 1 4 3 2 7
48 1 4 5 6 7
49 2 4 5 10 13
50 2 4 5 10 17
51 1 5 10 13 14
52 1 5 10 17 16
53 2 6 5 10 13
54 2 6 5 10 17
55 1 7 6 5 10
56 1 10 13 14 15
57 1 10 17 16 15
58 1 13 10 17 16
59 1 13 14 15 16
60 1 14 13 10 17
61 1 14 15 16 17
62 3 1 2 7 3
63 3 4 6 5 10
64 3 13 17 10 5
.
.
.

3. The build layer script

Source code of the 'build layer' shell script:

```
#!/bin/bash

echo 'data file:'

read file;

echo 'how many molecules in x: '
```

```
read num_mol_x;

echo 'how many molecules in y: '

read num_mol_y;

echo 'lattice constant'

read latt_cnst;

echo 'cover rate (0 - 100 %)'
read cover;

echo 'name of new data file'

read newdata

echo '(1) randomly dstritbued
(2) randomly distributed (constant seed)
(3) herringbone'

read choice

num_atoms='grep atoms $file | awk '{print $1}'';
num_bonds='grep bonds $file | awk '{print $1}'';
num_angles='grep angles $file | awk '{print $1}'';
num_dihedrals='grep dihedrals $file | awk '{print $1}'';
num_impropers='grep impropers $file | awk '{print $1}'';

n_atom_types='grep -e 'atom types' $file | awk '{print $1}'';
n_bond_types='grep -e 'bond types' $file | awk '{print $1}'';
n_angle_types='grep -e 'angle types' $file |awk '{print $1}'';
n_dihedral_types='grep -e 'dihedral types' $file | awk '{print $1}'';
n_improper_types='grep -e 'improper types' $file | awk '{print $1}'';
```

Appendix

```
atom_offset='expr $num_atoms + 1';

grep Atoms $file -A$atom_offset | grep -v Atoms > section_atoms.dat;

bond_offset='expr $num_bonds + 1';
grep Bonds $file -A$bond_offset | grep -v Bonds > section_bonds.dat;

angle_offset='expr $num_angles + 1';
grep Angles $file -A$angle_offset | grep -v Angles > section_angles.dat

dihedral_offset='expr $num_dihedrals + 1';
grep Dihedrals $file -A$dihedral_offset | grep -v Dihedrals > section_dihedrals.d

if test $choice -eq 3;
then
    /home/lukas/md_calc/utils/replicate_herringbone $num_atoms $num_bonds
    $num_angles $num_dihedrals $num_impropers $num_mol_x $num_mol_y
    $latt_cnst $cover ;
else
    /home/lukas/md_calc/utils/replicate $num_atoms $num_bonds $num_angles
    $num_dihedrals $num_impropers $num_mol_x $num_mol_y $latt_cnst
    $cover $choice ;
fi

num_mol='cat number_of_molecules.dat';
let num_atoms_new=$num_atoms*$num_mol;
let num_bonds_new=$num_bonds*$num_mol;
let num_angles_new=$num_angles*$num_mol;
let num_dihedrals_new=$num_dihedrals*$num_mol;
let num_impropers_new=$num_impropers*$num_mol;

let mass_offset=$n_atom_types+1;
let bond_coeff_offset=$n_bond_types+1;
```

```
let angle_coeff_offset=$n_angle_types+1;
let dihedral_coeff_offset=$n_dihedral_types+1;
let improper_coeff_offset=$n_improper_types+1;

echo 'LAMMPS data file generated from '$file> data.$newdata;
echo >> data.$newdata;
echo $num_atoms_new 'atoms' >> data.$newdata;
echo $num_bonds_new 'bonds' >> data.$newdata;
echo $num_angles_new 'angles' >> data.$newdata;
echo $num_dihedrals_new 'dihedrals' >> data.$newdata;
echo $num_impropers_new 'impropers' >> data.$newdata;
echo >> data.$newdata;
grep types $file >> data.$newdata;
echo >> data.$newdata;
cat xy_lim.dat >> data.$newdata;
echo '-50 150 zlo zhi' >> data.$newdata;
echo >> data.$newdata;
grep Masses $file -A$mass_offset >> data.$newdata;
echo >> data.$newdata;
grep -e 'Pair Coeffs' $file -A$mass_offset >> data.$newdata;
echo >> data.$newdata;
grep -e 'Bond Coeffs' $file -A$bond_coeff_offset >> data.$newdata;
echo >> data.$newdata;
grep -e 'Angle Coeffs' $file -A$angle_coeff_offset >> data.$newdata;
echo >> data.$newdata;
grep -e 'Dihedral Coeffs' $file -A$dihedral_coeff_offset >> data.$newdata;
echo >> data.$newdata;
grep -e 'Improper Coeffs' $file -A$improper_coeff_offset >> data.$newdata;
echo >> data.$newdata;
echo 'Atoms' >> data.$newdata;
echo >> data.$newdata;
cat atoms_section_new.dat >> data.$newdata;
echo >> data.$newdata;
echo 'Bonds' >> data.$newdata;
echo >> data.$newdata;
```


Appendix

```
cat bonds_section_new.dat >> data.$newdata;
echo >> data.$newdata;
echo 'Angles' >> data.$newdata;
echo >> data.$newdata;
cat angles_section_new.dat >> data.$newdata;
echo >> data.$newdata;
echo 'Dihedrals' >> data.$newdata;
echo >> data.$newdata;
cat dihedrals_section_new.dat >> data.$newdata;
```

The script 'build layer' given above requires the following C++ script called replicate:

```
#include "types.cpp"
```

```
MatDbl layer(MatDbl atoms_init, MatDbl lattice, int number_mol,
             int atoms);
```

```
MatInt bonds(MatInt bonds_init, int n_bonds, int n_molecules,
             int n_atoms);
```

```
MatInt angles(MatInt angles_init, int n_angles, int n_molecules,
             int n_atoms);
```

```
MatInt dihedrals(MatInt dihedrals_init, int n_dihedrals,
                int n_molecules, int n_atoms);
```

```
MatInt impropers(MatInt impropers_init, int n_impropers,
                int n_molecules, int n_atoms);
```

```
/* ##### MAIN #####*/
```

```
int main(int argc, char* argv[]){
```

```
    /* 1st arg: number of atoms per molecule
```

```
       2nd arg: number of bonds within the molecule
```

```
       3rd arg: number of angles within the molecule
```

```
       4th arg: number of dihedrals
```

```
       5th arg: number of impropers
```

```
       6th arg: desired number of molecules in x direction
```

3. The build layer script

```
7th arg: desired number of molecules in y direction
8th arg: lattice constant
9th arg: coverage
10th arg: choice of randomness
*/

int num_atoms, num_bonds, num_angles, num_dihedrals, num_improper,
    rate, rat_x , rat_y, molecules, choice;
double lattice_const, x_low, x_high, y_low, y_high, cover,
    surf_lat_cnst, x_space, y_space;

num_atoms = atoi(argv[1]);
num_bonds = atoi(argv[2]);
num_angles = atoi(argv[3]);
num_dihedrals = atoi(argv[4]);
num_improper = atoi(argv[5]);
rat_x = atoi(argv[6]);
rat_y = atoi(argv[7]);
lattice_const = atof(argv[8]);
cover = atof(argv[9]);
choice = atoi(argv[10]);

MatDbl sec_atoms(num_atoms,7),lattice(rat_x*rat_y,3);

std::fstream e1;
e1.open("section_atoms.dat", std::ios::in);
for(int k=0; k<num_atoms;k++){
    for(int l=0; l<7;l++){
        e1 >> sec_atoms(k,l);
    }
}
e1.close();

/*-----Lattice----- */
```

Appendix

```
surf_lat_cnst = lattice_const*sqrt(2)/2.0;
x_space = sqrt(3)*surf_lat_cnst;
y_space = 1.5*surf_lat_cnst;
cout <<lattice_const <<"    "<< surf_lat_cnst << endl;
cout << x_space << "    "<< y_space <<endl;
std::vector<int> index;
if(choice == 1){
    srand(time(NULL));
}
else if(choice == 2){
    srand(1297845);
}
int x_cnt=0, y_cnt=0;
for(int i=0 ; i < rat_x*rat_y; i++){
    lattice(i,0) = x_cnt*x_space+(x_space/2)*((y_cnt)%2);
    lattice(i,1) = y_cnt*y_space;
    lattice(i,2) = 0;
    x_cnt++;
    if((i+1)%rat_x == 0){
        y_cnt++;
        x_cnt=0;
    }

    if((rand()%10000+1)/100.0 < cover) index.push_back(i);
}
molecules = (int) index.size();
cout << "molecules "<< molecules <<endl;
MatDbl lattice_final(molecules,3);
int w;
for(int k=0; k<molecules; k++){
    w = index[k];
    lattice_final(k,0) = lattice(w,0);
    lattice_final(k,1) = lattice(w,1);
    lattice_final(k,2) = lattice(w,2);
}
```

```

std::fstream num;
num.open("number_of_molecules.dat", std::ios::out);
num << molecules << endl;
num.close();
/* ----- Atoms ----- */
MatDbl sec_atoms_new(molecules*num_atoms,7);
sec_atoms_new = layer(sec_atoms,lattice_final,molecules,num_atoms);
std::fstream e2;
e2.open("atoms_section_new.dat", std::ios::out);
  for (int g=0; g<molecules*num_atoms; g++){
    e2 << sec_atoms_new(g,0) << " " << sec_atoms_new(g,1) << " " <<
sec_atoms_new(g,2)
<< " " << sec_atoms_new(g,3) << " " << sec_atoms_new(g,4) <<
" " << sec_atoms_new(g,5)
<< " " << sec_atoms_new(g,6) <<endl;
  }
e2.close();

/* -----Bonds ----- */

MatInt sec_bonds(num_bonds,4), sec_bonds_new(molecules*num_bonds,4);

std::fstream f1;
f1.open("section_bonds.dat", std::ios::in);
for(int k=0; k<num_bonds; k++){
  for(int l=0; l<4; l++){
    f1 >> sec_bonds(k,l);
  }
}
f1.close();

sec_bonds_new = bonds(sec_bonds, num_bonds, molecules, num_atoms);

std::fstream f2;

```

Appendix

```
f2.open("bonds_section_new.dat", std::ios::out);
for(int k=0; k<molecules*num_bonds; k++){
    f2 << sec_bonds_new(k,0) << " " << sec_bonds_new(k,1) << " " <<
        sec_bonds_new(k,2) << " " << sec_bonds_new(k,3) << " " << endl;
}
f2.close();

/* -----Angles-----*/

MatInt sec_angles(num_angles,5), sec_angles_new(molecules*num_angles,5);

std::fstream an1;
an1.open("section_angles.dat", std::ios::in);

for(int k=0; k<num_angles; k++){
    for(int l=0; l<5; l++){
        an1 >> sec_angles(k,l);
    }
}
an1.close();

sec_angles_new = angles(sec_angles, num_angles, molecules, num_atoms);

std::fstream an2;
an2.open("angles_section_new.dat", std::ios::out);
for(int k=0; k<molecules*num_angles; k++){
    an2 << sec_angles_new(k,0) << " " << sec_angles_new(k,1) << " "
<< sec_angles_new(k,2) << " " << sec_angles_new(k,3) << " "
<< sec_angles_new(k,4) << endl;
}
an2.close();

/*-----Dihedrals-----*/
```

```

MatInt sec_dihedrals(num_dihedrals,6),
    sec_dihedrals_new(molecules*num_dihedrals,6);

std::fstream di1;
di1.open("section_dihedrals.dat",std::ios::in);
for(int k=0; k<num_dihedrals; k++){
    for(int l=0; l<6; l++){
        di1 >> sec_dihedrals(k,l);
    }
}
di1.close();

sec_dihedrals_new = dihedrals(sec_dihedrals, num_dihedrals, molecules,
num_atoms);

std::fstream di2;
di2.open("dihedrals_section_new.dat",std::ios::out);
for(int k=0; k<num_dihedrals*molecules; k++){
    di2 << sec_dihedrals_new(k,0) << " " << sec_dihedrals_new(k,1) << " "
<< sec_dihedrals_new(k,2) << " " << sec_dihedrals_new(k,3) << " "
<< sec_dihedrals_new(k,4) << " " << sec_dihedrals_new(k,5) << " "
<<endl;
}
di2.close();

/*-----computing simulation box boundaries ----- */

x_low = 0;
x_high = rat_x*sqrt(6)/2*lattice_const;

y_low = 0;
y_high = rat_y*0.75*sqrt(2)*lattice_const;

std::fstream lim;

```

Appendix

```
lim.open("xy_lim.dat", std::ios::out);
lim << x_low << " " << x_high << " xlo xhi" << endl;
lim << y_low << " " << y_high << " ylo yhi" << endl;
lim.close();

return 0;
}

/*#####END MAIN#####*/

/*#####
##### FUNCTIONS #####
#####
*/

MatDbl layer(MatDbl atoms_init, MatDbl lattice, int number_mol,
            int atoms){
    /* The function 'layer' reads the data from the 'Atoms' section
       of the data file (atoms_init) and generates a new 'Atoms' section,
       with number_mol molecules having randomized twist angles */

    double phi;
    double const pi=3.141592653589793;
    int count=0;
    MatDbl atoms_final(number_mol*atoms,7);

    srand(13395);

    for(int k=0; k<number_mol; k++){

        phi = (rand()%100+1)/100.0*pi;

        for(int l=0; l< atoms; l++){
```

```

    atoms_final(l+k*atoms,0) = atoms_init(l,0) + k*atoms;
    atoms_final(l+k*atoms,1) = k;
    atoms_final(l+k*atoms,2) = atoms_init(l,2);
    atoms_final(l+k*atoms,3) = atoms_init(l,3);
    atoms_final(l+k*atoms,4) = atoms_init(l,4)*cos(phi) -
atoms_init(l,5)*sin(phi)
+ lattice(k,0);
    atoms_final(l+k*atoms,5) = atoms_init(l,4)*sin(phi) +
atoms_init(l,5)*cos(phi)
+ lattice(k,1);
    atoms_final(l+k*atoms,6) = atoms_init(l,6) + lattice(k,2);
}
}
return atoms_final;
}

/* -----*/

MatInt bonds(MatInt bonds_init, int n_bonds, int n_molecules,
    int n_atoms){

    MatInt bonds_final(n_bonds*n_molecules,4);

    for(int j1=0; j1<n_bonds*n_molecules; j1++){
        for(int j2=0; j2<4; j2++){
            bonds_final(j1,j2) = 0;
        }
    }

    for(int k=0; k<n_molecules; k++){
        for(int l=0; l<n_bonds; l++){
            bonds_final(l+k*n_bonds,0) = bonds_init(l,0) + k*n_bonds;
            bonds_final(l+k*n_bonds,1) = bonds_init(l,1);

```


Appendix

```
        bonds_final(1+k*n_bonds,2) = bonds_init(1,2) + k*n_atoms;
        bonds_final(1+k*n_bonds,3) = bonds_init(1,3) + k*n_atoms;
    }
}
return bonds_final;
}

/* -----*/

MatInt angles(MatInt angles_init, int n_angles, int n_molecules,
              int n_atoms){

    MatInt angles_final(n_angles*n_molecules,5);

    for(int k=0; k<n_molecules; k++){
        for(int l=0; l<n_angles; l++){
            angles_final(1+k*n_angles,0) = angles_init(1,0) + k*n_angles;
            angles_final(1+k*n_angles,1) = angles_init(1,1);
            angles_final(1+k*n_angles,2) = angles_init(1,2) + k*n_atoms;
            angles_final(1+k*n_angles,3) = angles_init(1,3) + k*n_atoms;
            angles_final(1+k*n_angles,4) = angles_init(1,4) + k*n_atoms;
        }
    }

    return angles_final;
}

/* -----*/

MatInt dihedrals(MatInt dihedrals_init, int n_dihedrals,
                int n_molecules, int n_atoms){

    MatInt dihedrals_final(n_dihedrals*n_molecules,6);

    for(int k=0; k<n_molecules; k++){
```

3. The build layer script

```
for(int l=0; l<n_dihedrals; l++){
    dihedrals_final(l+k*n_dihedrals,0) = dihedrals_init(l,0) + k*n_dihedrals;
    dihedrals_final(l+k*n_dihedrals,1) = dihedrals_init(l,1);
    dihedrals_final(l+k*n_dihedrals,2) = dihedrals_init(l,2) + k*n_atoms;
    dihedrals_final(l+k*n_dihedrals,3) = dihedrals_init(l,3) + k*n_atoms;
    dihedrals_final(l+k*n_dihedrals,4) = dihedrals_init(l,4) + k*n_atoms;
    dihedrals_final(l+k*n_dihedrals,5) = dihedrals_init(l,5) + k*n_atoms;
}
}

return dihedrals_final;
}

/*-----*/

MatInt impropers(MatInt impropers_init, int n_impropers,
int n_molecules, int n_atoms){

MatInt impropers_final(n_impropers*n_molecules,6);

for(int k=0; k<n_molecules; k++){
    for(int l=0; l<n_impropers; l++){
        impropers_final(l+k*n_impropers,0) = impropers_init(l,0) + k*n_impropers;
        impropers_final(l+k*n_impropers,1) = impropers_init(l,1);
        impropers_final(l+k*n_impropers,2) = impropers_init(l,2) + k*n_atoms;
        impropers_final(l+k*n_impropers,3) = impropers_init(l,3) + k*n_atoms;
        impropers_final(l+k*n_impropers,4) = impropers_init(l,4) + k*n_atoms;
        impropers_final(l+k*n_impropers,5) = impropers_init(l,5) + k*n_atoms;
    }
}
return impropers_final;
}
```

4. MATLAB codes

4.1. Angles

The source code of the MATLAB script for computing the angles and angle distributions.

```
clear all
close all

x_max = 74.92;
y_max = 77.86;
num_steps = 101; %number of recorded time steps
%load coordinate file of initial configuraion
coords = load('layer.au_bpt_run2_nve_300_0');
num_mol = max(coords(:,2)) + 1; % extract number of molecules

tilt_angle = zeros(num_steps,num_mol);
tw_angle = zeros(num_steps,num_mol);
az_angle = zeros(num_steps,num_mol);
tw_alt = zeros(num_steps,num_mol);

%vectors for histograms
sp_twt = 0:.1:90;
sp_tw = 0:0.5:180;
sp_az = 0:0.5:360;

S_dock_coords = zeros(num_mol,5,num_steps);

cn = zeros(4,3);
cn_p = zeros(4,3);
cn_pabs = zeros(1,4);
m_tw = zeros(1,2*num_steps);
for l = 1:num_steps %loop over all recorded timesteps

    %loading coordinate file for current timestep
```

```

coords = load(['layer.au_bpt_run2_nve_300_',num2str(1-1)]);

for k = 1:num_mol % loop over all molecules

% extracting molecular coordinates for the currently considered molecule

    mol_ind = find(coords(:,2)==k-1);
    mol_coords = coords(mol_ind,:);
    mol_coords = sortrows(mol_coords);

%extracting the sulfur coordinates (here atom type 1)
    S_dock_ind = find(mol_coords(:,3) ==1);
    S_dock_coords(k,:,1) = mol_coords(S_dock_ind,2:6);
%extracting the carbon coordinates (here atom type 2)
    C_ind = find(mol_coords(:,3) ==2);
    C_coords = mol_coords(C_ind,2:6);

%calculating connection vector between docking atom and the very top C atom
    sc1 =connectpoint(C_coords(1,3:5),S_dock_coords(k,3:5,1),x_max,y_max);
    sc1_abs = sqrt(sum(sc1.^2));

%calculating connection vector between C atom at the very bottom and the very top
    c =connectpoint(C_coords(4,3:5),C_coords(1,3:5),x_max,y_max);
    c_abs = (c(1)^2+c(2)^2+c(3)^2)^0.5;

%computing an averaged vector of the two vectors above
    ax = mean([sc1;c],1);
    ax_abs = sqrt(sum(ax.^2));

%TILT ANGLE

    tilt_angle(1,k) = acos(ax(3)/(ax_abs))./(pi)*180;

% TWIST ANGLE

%calculating connection vector between several C atoms of a molecule
    cl_1 = connectpoint(C_coords(1,3:5),C_coords(2,3:5),x_max,y_max);
    cl_2 = connectpoint(C_coords(1,3:5),C_coords(3,3:5),x_max,y_max);

```

Appendix

```
cl_3 = connectpoint(C_coords(1,3:5),C_coords(4,3:5),x_max,y_max);  
cl_4 = connectpoint(C_coords(1,3:5),C_coords(5,3:5),x_max,y_max);  
cl_5 = connectpoint(C_coords(1,3:5),C_coords(6,3:5),x_max,y_max);
```

```
cl = [cl_1;cl_2;cl_3;cl_4;cl_5];
```

```
%calculating the normal vectors of the connection vectors
```

```
for cr = 1:4
```

```
cn(cr,:) = cross(cl(1,:),cl(cr+1,:));
```

```
cn_p(cr,:) = [cn(cr,1:2), 0];
```

```
cn_pabs(cr) = sqrt(sum(cn_p(cr,:).^2));
```

```
end
```

```
%averaging over the normal vectors
```

```
cn_pa = mean(cn_p,1);
```

```
cn_pabsa=mean(cn_pabs);
```

```
tw_angle(1,k) = acos(cn_pa(2)/cn_pabsa)./(pi)*180;
```

```
if cn_pa(1) < 0 && cn_pa(2) < 0
```

```
tw_angle(1,k) = 270 - tw_angle(1,k);
```

```
elseif cn_pa(1) >= 0 && cn_pa(2) >= 0
```

```
tw_angle(1,k) = tw_angle(1,k) + 90;
```

```
elseif cn_pa(1) >= 0 && cn_pa(2) < 0
```

```
tw_angle(1,k) = tw_angle(1,k) - 90;
```

```
elseif cn_pa(1) < 0 && cn_pa(2) >= 0
```

```
tw_angle(1,k) = 90 - tw_angle(1,k);
```

```
end
```

```
% AZIMUTHAL ANGLE
```

```
mpl = [ax(1:2), 0]; %projection of the molecular axis on x-y plane
```

```

mpl_abs = sqrt(sum(mpl.^2));

az_angle(1,k) = acos(mpl(2)/mpl_abs)./pi*180;

if mpl(1) < 0 && mpl(2) < 0
    az_angle(1,k) = 360 - az_angle(1,k);

elseif mpl(1) >= 0 && mpl(2) >= 0
    az_angle(1,k) = az_angle(1,k) ;

elseif mpl(1) >= 0 && mpl(2) < 0
    az_angle(1,k) = az_angle(1,k);

elseif mpl(1) < 0 && mpl(2) >= 0
    az_angle(1,k) = -az_angle(1,k);
end
end
end

%calculating the time averaged angles
T = repmat(1:num_steps,num_mol,1);
avc_tilt_angle = cumsum(tilt_angle,1)./T.';
avc_twist_angle = cumsum(tw_angle,1)./T.';
avc_az_angle = cumsum(az_angle,1)./T.';

mavc_tilt = mean(avc_tilt_angle,2);

[c_hi] = find(avc_twist_angle(1,:) >=90);
[c_lo] = find(avc_twist_angle(1,:) < 90);

[c_hi_a] = find(tw_alt(1,:) >=90);
[c_lo_a] = find(tw_alt(1,:) < 90);

```

Appendix

```
avc_twist_lo = avc_twist_angle(:,c_lo);
avc_twist_hi = avc_twist_angle(:,c_hi);
mavc_twist_lo = mean(avc_twist_lo,2);
mavc_twist_hi = mean(avc_twist_hi,2);

tilt_angle_t = mean(tilt_angle(:,:),1);
az_angle_t = mean(az_angle(:,:),1);
tw_angle_t = mean(tw_angle(:,:),1);

tilt_angle_m = mean(tilt_angle(:,:),2);
az_angle_m = mean(az_angle(:,:),2);
tw_angle_m_lo = mean(tw_angle(:,c_lo),2);
tw_angle_m_hi = mean(tw_angle(:,c_hi),2);

std_tilt = std(tilt_angle(:,:),0,2);
std_tw_lo = std(tw_angle(:,c_lo),0,2);
std_tw_hi = std(tw_angle(:,c_hi),0,2);
std_az = std(az_angle(:,:),0,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% graphic output
scrsz = get(0,'ScreenSize');
% distribution tilt angel

[d1,s1] = histc(tilt_angle',sp_twl);
hd1 = mean(d1./num_mol./0.1,2);
options = optimset('TolFun',1e-8,'Display','off');
[tlt_fit] = ...
    lsqcurvefit(@mygf,[1 12 5],sp_twl,hd1',[],[],options);

figure(1)
f1 = gcf;
```

```

set(f1,'Position',...
     [1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2],'Name','tilt angle')

hold on
bart1 = bar(sp_twt,hd1);
set(bart1,'LineStyle','none','FaceColor','red')
plot(0:0.01:30,mygf(tlt_fit,0:0.01:30))
xlim([-1 90]);
hold off
xlabel('degrees','FontSize',20,'FontWeight','bold')
ylabel('p','FontSize',20,'FontWeight','bold');
ax1 = gca;
set(ax1,'LineWidth',3,'FontSize',20,'FontWeight','bold')

% distribution twist angle

[ae1,q1] = histc(tw_angle',sp_tw);
htw_av=mean(ae1./num_mol./5,2);
htw1_av=zeros(size(htw_av));
htw1_av(sp_tw<90) = htw_av(sp_tw<90);

htw2_av = zeros(size(htw_av));
htw2_av(sp_tw>=90) = htw_av(sp_tw>=90);
[tw1_fit] = ...
    lsqcurvefit(@mygf,[1 50 10],sp_tw,htw1_av',[],[],options);
[tw2_fit] = ...
    lsqcurvefit(@mygf,[1 130 10],sp_tw,htw2_av',[],[],options);

figure(2)
f2 = gcf;
set(f2,'Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2],'Name','twist angle')
hold on

bar1 = bar(sp_tw,htw_av);

```


Appendix

```
set(baraz1,'LineStyle','none','FaceColor','red')
plot(0:0.01:90,mygf(tw1_fit,0:0.01:90))
plot(90:0.01:180,mygf(tw2_fit,90:0.01:180))
xlim([0 180]);
hold off
xlabel('degrees','FontSize',20,'FontWeight','bold')
ylabel('p','FontSize',20,'FontWeight','bold');

ax1 = gca;
set(ax1,'LineWidth',3,'FontSize',20,'FontWeight','bold')

% distribution azimuthal angle

[az1,az2] = histc(az_angle',sp_az);
haz_av = mean(az1./num_mol./5,2);
plv_az= 90:0.01:270;
[az_fit] = ...
    lsqcurvefit(@mygf,[1 180 5],sp_az,haz_av',[],[],options);

figure(3)
f3 = gcf;
set(f3,'Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2],'Name','azimutangle')
hold on
baraz1 = bar(sp_az,haz_av);
set(baraz1,'LineStyle','none','FaceColor','red')
plot(plv_az,mygf(az_fit,plv_az))
xlim([min(sp_az) max(sp_az)]);
hold off
xlabel('degrees','FontSize',20,'FontWeight','bold')
ylabel('p','FontSize',20,'FontWeight','bold');
ax1 = gca;
set(ax1,'LineWidth',3,'FontSize',20,'FontWeight','bold')

% write data
```

```

file1 = fopen('fit_parameters_bpt_r2_nve_300.dat','w');
fprintf(file1,...
' %6.4f %6.4f %6.4f\n %6.4f %6.4f %6.4f\n %6.4f %6.4f %6.4f\n %6.4f %6.4f %6.4f\n', .
tlt_fit(1),tlt_fit(2),tlt_fit(3),tw1_fit(1), tw1_fit(2), tw1_fit(3), ...
tw2_fit(1), tw2_fit(2), tw2_fit(3), az_fit(1), az_fit(2),az_fit(3));
fclose(file1);

%%
% evolution of the average angles in time

figure(4)
f4 = gcf;
set(f4,'Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2])
subplot(4,1,1)
hold on
plot(tilt_angle_m,'.')
ylim([0 25]);
hold off

subplot(4,1,2)
hold on
plot(tw_angle_m_hi,'.')
ylim([90 180]);
hold off

subplot(4,1,3)
hold on
plot(tw_angle_m_lo,'.')
ylim([0 90]);
hold off

subplot(4,1,4)
hold on
plot(az_angle_m,'.')

```

Appendix

```
ylim([min(az_angle_m)-5 max(az_angle_m)+5]);
hold off

% standard deviations

figure(5)
f5 = gcf;
set(f5,'Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2])
subplot(4,1,1)
plot(std_tilt, '.');

subplot(4,1,2)
plot(std_tw_lo, '.');

subplot(4,1,3)
plot(std_tw_hi, '.');

subplot(4,1,4)
plot(std_az, '.');

% cumulative averages

figure(6)
f6 = gcf;
set(f6,'Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2])

subplot(4,1,1)
hold on
plot(avc_tilt_angle)
hold off

subplot(4,1,2)
plot(avc_twist_hi)
```

```
subplot(4,1,3)
plot(avc_twist_lo)

subplot(4,1,4)
plot(avc_az_angle);
```

4.2. Displacements

Source codes of the MATLAB script used for the displacement analysis. Files with coordinates of the fcc-hollow, hcp-hollow and top sites throughout the simulation cell, coordinate files of the monolayer molecules and a file with the mean square displacements are needed for input.

```
clear all;
close all;

num_mol=270;
num_steps=101;
x_max = 74.92;
y_max = 77.86;

f1 = fopen('msd.dock_au_bpt_bcc_run2_eq_nve_300');
m300 = textscan(f1,'%15.13f %15.13f %15.13f %15.13f %15.13f','CommentStyle','#');
fclose(f1);
msd_300 = [m300{1},m300{2},m300{3},m300{4},m300{5}];
t1=linspace(0,500,length(msd_300));

figure(1)
hold on
plot(t1,msd_300(:,5),'.-')
xlabel('t [ps]');
ylabel('msd');
hold off

lattice_const = 4.0782;
surf_lat_cnst = lattice_const*sqrt(2)/2.0;
```

Appendix

```
%loading files with fcc, hcp and top coordinates of the simulation cell
f1 = fopen('fcc_sites.lampstr');
fcc = textscan(f1,'%12.8f %12.8f %12.8f','HeaderLines',9);
fclose(f1);
sites_fcc=[fcc{1},fcc{2}];
f1 = fopen('top_sites.lampstr');
top = textscan(f1,'%12.8f %12.8f %12.8f','HeaderLines',9);
fclose(f1);
sites_top=[top{1},top{2}];

f1 = fopen('hcp_sites.lampstr');
hcp = textscan(f1,'%12.8f %12.8f %12.8f','HeaderLines',9);
fclose(f1);
sites_hcp=[hcp{1},hcp{2}];
br=length(sites_fcc);
S_dock_coords = zeros(num_mol,6,num_steps);
S_d=zeros(num_mol,3,num_steps);

phi=linspace(0,2*pi,1000);

r1=sqrt(3)/6*surf_lat_cnst;

c_fcc=[.2 .2 .2];
c_hcp = [.5 .5 .5];
c_top = [.7 .7 .7];

% calculating the displacements of the docking atoms
for l=1:num_steps
    coords = load(['layer.au_bpt_run2_nve_300_',num2str(l-1)]);
    for k=1:num_mol

        mol_ind = find(coords(:,2)==k-1);
        mol_coords = coords(mol_ind,:);
        s_ind=find(mol_coords(:,3)==1);
```

```

S_dock_coords(k,:,l) = mol_coords(s_ind,:);

S_d(k,:,l)= connectpoint(S_dock_coords(k,4:6,l),S_dock_coords(k,4:6,1),x_max,y_max);
end
end

figure(6) %illustration of the different docking sites

hold on
plot(0,0,'rp','Markersize',15,'MarkerFaceColor','r');
plot(-0.7209,-0.4164,'rh','Markersize',15,'MarkerFaceColor','r')
plot(1.4419,0.8325,'rd','Markersize',15,'MarkerFaceColor','r')
plot(-1.4419,-0.8325,'ro','Markersize',15,'MarkerFaceColor','r')
legend('fcc','bridge','top','hcp')
%fcc
fill(0+r1*sin(phi),0+r1*cos(phi),c_fcc,'LineStyle','none');
fill(2.8837+r1*sin(phi),0+r1*cos(phi),c_fcc,'LineStyle','none');
fill(-2.8837+r1*sin(phi),0+r1*cos(phi),c_fcc,'LineStyle','none');
fill(1.4419+r1*sin(phi),3*0.8325+r1*cos(phi),c_fcc,'LineStyle','none');
fill(-1.4419+r1*sin(phi),3*0.8325+r1*cos(phi),c_fcc,'LineStyle','none');
fill(-1.4419+r1*sin(phi),-3*0.8325+r1*cos(phi),c_fcc,'LineStyle','none');
fill(1.4419+r1*sin(phi),3*0.8325+r1*cos(phi),c_fcc,'LineStyle','none');
fill(1.4419+r1*sin(phi),-3*0.8325+r1*cos(phi),c_fcc,'LineStyle','none');
%hcp
fill(-1.4419+r1*sin(phi),-0.8325+r1*cos(phi),c_hcp,'LineStyle','none')
fill(1.4419+r1*sin(phi),-0.8325+r1*cos(phi),c_hcp,'LineStyle','none')
fill(0+r1*sin(phi),1.6649+r1*cos(phi),c_hcp,'LineStyle','none')
fill(-2.8837+r1*sin(phi),1.6649+r1*cos(phi),c_hcp,'LineStyle','none');
fill(2.8837+r1*sin(phi),1.6649+r1*cos(phi),c_hcp,'LineStyle','none');
%top
fill(1.4419+r1*sin(phi),0.8325+r1*cos(phi),c_top,'LineStyle','none')
fill(-1.4419+r1*sin(phi),0.8325+r1*cos(phi),c_top,'LineStyle','none')
fill(0+r1*sin(phi),-1.6649+r1*cos(phi),c_top,'LineStyle','none')
fill(2.8837+r1*sin(phi),-1.6649+r1*cos(phi),c_top,'LineStyle','none');
fill(-2.8837+r1*sin(phi),-1.6649+r1*cos(phi),c_top,'LineStyle','none');

```

Appendix

```
plot(0,0,'rp','Markersize',15,'MarkerFaceColor','r');
plot(-0.7209,-0.4164,'rh','Markersize',15,'MarkerFaceColor','r')
plot(1.4419,0.8325,'rd','Markersize',15,'MarkerFaceColor','r')
plot(-1.4419,-0.8325,'ro','Markersize',15,'MarkerFaceColor','r')
axis equal
axis off;
    hold off

ts=[1,25,50,75,101]; %considered time steps
pos_cnt=zeros(4,length(ts)); %line1: fcc, line2: hcp, line3: top; line4: bridge
sw=1;

for ff=1:length(ts) %loop over considered time steps
    ti = ts(ff);

    for k=1:num_mol %loop over all molecules

        for g1=1:br %loop over all fcc docking sites
            ri1 = connectpoint(S_dock_coords(k,4:5,ti),sites_fcc(g1,:),x_max,y_max);
            rd = sqrt(sum(ri1.^2));
            if rd < r1
                pos_cnt(1,ff)=pos_cnt(1,ff)+1;
                sw=0;
                break
            end
        end

        if sw == 1
            for g1=1:br %loop over all hcp docking sites
                ri1 = connectpoint(S_dock_coords(k,4:5,ti),sites_hcp(g1,:),x_max,y_max);
                rd = sqrt(sum(ri1.^2));
                if rd < r1
                    pos_cnt(2,ff)=pos_cnt(2,ff)+1;
                    sw=0;
                end
            end
        end
    end
end
```

```

                break
            end
        end
    end
    if sw == 1
        for g1=1:br %loop over all top docking sites

            ri1 = connectpoint(S_dock_coords(k,4:5,ti),sites_top(g1,:),x_max,y_max);
            rd = sqrt(sum(ri1.^2));
            if rd < r1
                pos_cnt(3,ff)=pos_cnt(3,ff)+1;
                sw=0;
                break
            end
        end
        if sw == 1
            pos_cnt(4,ff)=pos_cnt(4,ff)+1;
        end
    end
end
end
sw=1;
end
end

```

```

dpl_s(:,:)=sqrt(sum(S_d(:,1:2,:).^2,2)); %calculate displacements for all molecules

```

```

sp_dpl=0:0.05:5;

```

```

[h,r] = histc(dpl_s(:,:),sp_dpl);

```

```

h = mean(h./241,2);

```

```

figure(3) %histogram of the displacements

```

```

fh3=gcf;

```


Appendix

```
set(fh3,'Name','xy displacement')
hold on
bar(sp_dpl,h,'r','LineStyle','none')

line([.8325 .8325], [0 max(h)/2],'LineStyle',':')
line([1.6649 1.6649], [0, max(h)/2],'LineStyle',':')
line([0 0], [0, max(h)/2],'LineStyle',':')
text(-.1,max(h)/2,'fcc')
text(.78,max(h)/2,'bridge')
text(1.7,max(h)/2,'hcp')
hold off

figure(5) %histogram of the population of the docking sites at the considered time st
hold on
bar(pos_cnt(:,:)./num_mol)
set(gca,'XTick',[1 2 3 4],'XTickLabel',{'fcc';'hcp';'top';'bridge'});
legend('0 ps', '25 ps', '50 ps', '75 ps', '100 ps')
hold off
```