Structural Relaxation of Molecular Angles under Pressure

MASTERARBEIT

zur Erlangung des akademischen Grades
Master of Science (M. Sc.)
im Fach Physik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
Institut für Physik
Humboldt-Universität zu Berlin

von
Herrn B. Sc. Eric Pierschel
geboren am 20.04.1990 in Berlin

Gutachter:

1. *Prof. Dr. Dr. h.c. Claudia Draxl*
2. *Prof. Dr. Stefan Kowarik*

eingereicht am: *31. July 2018*

Abstract

To predict material properties, such as strength or conductivity, precise calculations are necessary in material research. The properties of a crystal depend strongly on the configuration of the atoms. A general atomic configuration is not necessarily the natural ("equilibrium") configuration of a crytal, which is the one corresponding to the global minimum of its total energy. In particular, for molecular crystals, the forces between atoms belonging to a single molecule are significantly stronger than the ones between atoms in different molecules. The diffenrence in scale of the forces makes the numerical determination of the orientation of molecules with respect to each other very difficult. For molecular crystals, the energy landscape is so complex that many configurations can be local minima of the total energy with relatively tiny energy differences. In this work, a new approach for the search of the energy minimum of a molecular crystal is introduced. A coordinate transformation is carried out to convert the atomic degrees of freedom, usually simple Cartesian coordinates of the atoms, to the molecular ones. The procedure allows for performing the optimization by considering either each molecule as a rigid rotator or with the full molecular degrees of freedom. These approaches are used for relaxing molecular crystals, like the oligo-acens, under pressure. *Ab-initio* total energies for each configuration are calculated using density-functional theory and including van-der Waals interactions as implemented in the full-potential all-electron code **`exciting`**.

Zusammenfassung

Materialeigenschaften wie Festigkeit oder Leitfähigkeit sind in der Materialforschung auf präzise Rechnungen angewiesen. Diese Eigenschaften eines Kristalls hängen implizit von der Konfiguration der Atome im Kristallgitter ab, auch wenn das Kristallgitter als statisch angesehen wird. Eine spezielle atomare Konfiguration ist nicht notwendigerweise die natürliche Gleichgewichtskonfiguration eines Kristalls, welche dem globalen Minimum seiner Gesamtenergie entspricht. Insbesondere für Molekülkristalle sind die Kräfte zwischen den Atomen, die zu einem einzelnen Molekül gehören, deutlich stärker als die zwischen unterschiedlichen Molekülen. Die unterschiedlichen Größenordnungen der Kräfte macht die numerische Bestimmung der Ausrichtung der Moleküle zueinander sehr schwierig. Die Energielandschaft ist in diesen Fällen so komplex dass viele lokale Energieminima mit relativ kleinen differenzen existieren können. In dieser Arbeit wird eine neue Methode zur Bestimmung von lokalen Energieminima eines Molekülkristalls vorgestellt. Dafür wird eine Koordinatentransformation durchgeführt, um die atomaren Freiheitsgrade, einfache kartesische Koordinaten der Atome, zu den molekularen Freiheitsgraden zu transformieren. Einerseits können die Molekühle als starre Rotatoren angesehen werden und andererseits können die vollen molekularen Freiheitsgrade berücksichtig werden. Mit diesen Ansätzen wurden Molekühlkristalle aus den Oligo-Acene Klassen unter Druck relaxiert. Für die Berechnungen wurde die Dichtefunktionaltheorie mit van-der Waals korekturen benutzt, wie sie in dem Paket **`exciting`** implementiert ist.

Contents

List of Figures

Motivation

The knowledge of materials and their properties is a necessary part of today's world. Every application requires engineers to decide which material is best suited to their particular problem [1].

Computer chips require special electronic properties [2]. Airplanes and cars but also buildings require physical properties such as strength or elasticity. Chemistry labs want vessels or gloves that can resist special acids or bases. If materials are toxic they must be shielded when used or there are required non-toxic alternatives. X-ray microscopes as well as telescopes need different optical properties. And there are many more examples of material requirements in the world.

To select an optional material for a special purpose is not an easy task regarding infinite of possible materials. To simplify this, the theoretical investigation and comparison of a large number of materials is very helpful. Several databases containing results of computer simulations of materials have been developed in recent years. Among these, the largest one developed by the NOMAD project [3], where numerical data on known material properties are stored.

Many of the materials to be investigated are crystalline and they are commonly defined by their periodicity [4]. This periodicity can be described by space groups. Thus, a three-dimensional crystal can be generated by periodically repeating a "small" part of it called unit cell. The space-covering displacement vectors of the unit cell are called lattice vectors. The structure can now be defined by the unit cell and its containing atoms.

Molecular crystals consist of two substructures with different scale of interaction forces [5]. On the one hand, the molecules consists of bounded atoms which form the shape of the structure of the molecules. Bonds between these atoms are relatively strong. On the other hand, the molecule arranged in the lattice which form the cystalline structure. Due to the larger distance between atoms in different molecules, the interaction forces, generally of the type of van-der-Waals forces, are relatively weak in this case. However, they are very important for the arrangement of the molecules in the crystal.

Starting a simulation, a computer code needs as input the atomic configuration of the considered crystal. However, the initial configuration does not need to correspond to the configuration found in nature. This is because the starting structure is often specified by hand or by experimental data, when available. But, to obtain suitable results of the system, the configuration with the minimum total energy has to be used. Therefore, all crystal structures which are investigated have to be optimized in terms of their energy and, then, all other desired properties can be determined. For simple crystals with only a few degrees of freedom the optimization is a relatively easy task. However, for molecular crystals, the task is highly complex. Firstly, because of the usually larger number of degrees of freedom of the system, and, secondly, due to the presence of weak intermolecular forces. Because of this, often several local minima of the energy are present, with only very low difference in the minimum values.

This work presents two new methods for the relaxation of molecular crystals by the introduction of a coordinate transformation from the simple Cartesian coordinates of the atoms to the molecular degrees of freedom. While the first method reduces the degrees of freedom of a molecule by considering the molecules as rigid rotators, the second method allows for using the full molecular degrees of freedom by introducing pseudo vibrational modes. These algorithms are tested for the anthracene molecular crystal at different pressures.

The structure of the work is the following: First, the theoretical basics is presented in Chapter 1. Then, the new methods for the energy optimization and their implementation is described in Chapter 2. Finally, results of the application of the new optimization methods are presented in Chapter 3.

Chapter 1

Structure Optimization

The determination, of which atomic configuration of a crystal has the lowest energy, is a very important step in the theoretical description of materials. This procedure is known under the name of "structural optimization" or sometimes "relaxation" due to the fact, that the task is finding the equilibrium state of the crystal.

In this chapter, firstly, an introduction is given on the mathematical and numerical concepts and algorithms which are related to the optimization of a general function. Secondly, the specific function that has to be minimized, here the total energy of a crystal, is introduced together with the theoretical description of how this energy can be calculated. Finally, the system is shown on which the test for the structural optimization will be performed.

1.1 Optimization Algorithms

In order to systematically approach the optimization, the general case is considered in which the extreme points of a given function need to be calculated. A necessary criterion for calculating the extreme points is that all derivatives of the function at this point become zero. If the derivatives of the function can be calculated analytically or numerically, a search for the zeros of the derivative function is required. In general, however, the function may not be exactly known analytically or is highly complex. Thus, there are problems where the direct analytical calculation of the derivative is impossible. In these cases, an iterative search algorithm can be used to approach the extremal point. To do this, often a starting point is required for the algorithm.

To describe the search algorithms, the mathematical formalism of Ref. [6] is used here. It is assumed that $f(\vec{X})$ is the function, whose extreme points $\vec{X}_e$ are searched for. Then, D from $\vec{X} \in \mathbb{R}^D$ is the dimension of the function values. This

also defines the number of degrees of freedom, which are involved in the optimization problem. For iterative algorithms the index k is used for the k-th iteration. In this way $f_k = f(\vec{X}_k)$ is the function value at the k-th iteration. The first derivative of the function f is called gradient $\vec{g}(\vec{X}) = \pm\vec{\partial} f(\vec{X})$. The sign depends on, whether a minimum or a maximum is searched for. Due to the fact that real physical systems tend to adopt the configuration with the minimum possible energy, only minima are considered in this work. $\hat{B}(\vec{X}) = \vec{\partial}\vec{\partial}^{\mathrm{T}} f(\vec{X})$ indicates the matrix of the second derivatives of the function f and is called Hessian matrix. Some algorithms only use the gradient values $\vec{g}_k = \vec{\partial} f(\vec{X}_k) = \vec{\partial} f_k$ or Hessian values $\hat{B}_k$ determined in a specific point $\vec{X}_k$.

A necessary criterion for an extreme point of a function is generally that the derivatives of the function at this point becomes zero. Thus, a numerical solution of the equation $\vec{g}(\vec{X}) = 0$ must to be found. A typically numerical algorithm for this search works as follows:

- Start at a certain (possibly randomly selected) point $\vec{X}_0$.
- Suggest a new point $\vec{X}_{k=1}$, which is closer to the solution.
- Repeat $\vec{X}_k \to \vec{X}_{k+1}$ until the desired accuracy is achieved.

The easiest way is always to suggest the next point in the direction of the greatest possible slope $\vec{g}_k = -\vec{\partial} f(\vec{X}_k)$ of the function. Some algorithms collect at each point as much information about the local environment of the function as possible in order to make the best possible prediction about the next point.

There are two main strategies for reaching the extreme point: Line search and trust-region search, which will be explained shortly in the following. However, both methods determine only local minima and not necessarily the global extrema of the function f. Typically, the movement is done along the gradients and stops at the next local minimum found. Using these methods, the iteration steps can not overcome the potential barriers present between all local minima. To find the global extreme, *e.g.*, a Monte Carlo search [7] may be used.

1.1.1 Line Search

A line-search algorithm selects a direction $\vec{p}_k$ from a starting point $\vec{X}_k$ and searches for lower function values $f(\vec{X}_{k+1})$ on this line. The suggested direction $\vec{p}_k$ does not necessarily have to be normalized and some information can be stored in the length of this vector, *e.g.*, how far is the point from an expected minimum.

The step length A_k of the algorithm is determined formally by the solution of $\min f(\vec{X}_k + A\,\vec{p}_k)$. In this way, the dimensionality of the function $\tilde{f}(A) = f(\vec{X}_k + A\,\vec{p}_k)$

is reduced to one. This procedure is repeated with new search directions $\vec{p}_k$ until a predetermined accuracy is reached.

An algorithm which allows for finding a good approximation of the minimum along a line is the backtracking line search based on the Armijo–Goldstein condition [8]. In this algorithm, two controlling parameters $\tau \in (0,1)$ and $c \in (0,1)$ and one line-search iteration parameter j are introduced. Starting with $j = 0$, j is incrementally increased and the step length is set to $A_j = \tau A_{j-1}$ until the following Armijo–Goldstein condition is fulfilled:

$$f(\vec{X}) - f(\vec{X} + A_j \vec{p}_k) \geq -A_j \, \tau \, c \, \vec{p}_k \cdot \vec{g}_k \; . \tag{1.1.1}$$

This means that the search length A_j starts at a maximum value and is reduced by τ until the condition is met. The variable c controls the break condition.

For the search direction, the negative gradient $\vec{p}_k = -\vec{\partial} f(\vec{X}_k)$ is usually used since this is the direction in which f falls the most. This method is called steepest-descent method. Nevertheless, in principle any other direction from any another algorithm can be used, even random directions.

1.1.2 Trust-Region Method

Another way to determine the extreme point is the trust-region search. In this search, the information obtained about f is put into a model function m_k, which describes the function near the investigated point $\vec{X}_k$. Usually, this function is a second-order Taylor expansion around the point $\vec{X}_k$:

$$m_k(\vec{X}_k + \vec{p}_k) = f_k + \vec{p}_k^{\,\mathrm{T}} \cdot \vec{\partial} f_k(\vec{X}_k) + \frac{1}{2} \vec{p}_k^{\,\mathrm{T}} \cdot \hat{B} \cdot \vec{p}_k \; . \tag{1.1.2}$$

The next search point $\vec{X}_{k+1}$ is now determined by the minimum of the model function m_k. However, using this method the next point has to be within a self defined "`trust region`". This is the domain where the model function m_k is still a good approximation to the original function f around the local point $\vec{X}_k$. If the minimum of the model function is outside of the trust region, then the displacement of the next proposed point $\vec{X}_{k+1}$ must be reduced until the suggested point is in the accepted range. The model function m_k is now iteratively recalculated until the changes fall below a predetermined limit.

The disadvantage of this search is that the exact second derivative of the function f is required. And if the searched extreme point is very far from the start value, the proposed new points always are at the edge of the trust region, which greatly reduces the convergence.

1.1.3 Newton Direction

The Newton-direction method is another way to determine the direction $\vec{p}_k$ of the local minimum of a line search. It may replace the steepest-descent method in a line search but it does not deliver information about how far to go in this direction. However, the norm of the vector $|\vec{p}_k|$ still depends on the gradient, indicating the distance to a minimum. It is also determined by the second-order Taylor expansion:

$$f(\vec{X}_k + \vec{p}_k) \approx f_k + \vec{p}_k^{\mathrm{T}} \cdot \vec{\partial} f + \frac{1}{2}\, \vec{p}_k^{\mathrm{T}} \cdot \hat{B} \cdot \vec{p}_k \; . \tag{1.1.3}$$

Then, the Newton direction becomes:

$$\vec{p}_k = -(\vec{\partial}\vec{\partial}^{\mathrm{T}} f)^{-1} \cdot \vec{\partial} f = -\hat{B}^{-1} \cdot \vec{\partial} f \tag{1.1.4}$$

$$(\vec{\partial} f)^{\mathrm{T}} \cdot \vec{p}_k = -\vec{p}_k^{\mathrm{T}} \cdot \hat{B} \cdot \vec{p}_k \; . \tag{1.1.5}$$

This direction $\vec{p}_k$ can be used, *e.g.*, for a line search. This procedure typically delivers a quadratic convergence [6]. The drawback of this method is that an exact Hessian matrix $\hat{B}$ is required.

1.1.4 Quasi-Newton Search

To overcome the disadvantage of the need of an exact Hessian matrix $\hat{B}$ a quasi-Newton search can be used. Quasi-Newton methods do not use the exact Hessian matrix, but an approximate matrix $\hat{B}_k$ at each single step k. In this way, the approximate matrix gets more and more information about the function to be minimized at each step. In general, the derivation can be written as follows:

$$\vec{\partial} f(\vec{X} + \vec{p}) = \vec{\partial} f(\vec{X}) + \vec{\partial}\vec{\partial}^{\mathrm{T}} f(\vec{X}) \cdot \vec{p} + ... \tag{1.1.6}$$

Assuming for each iteration step $\vec{X} \to \vec{X}_k$ and $\vec{p} \to \vec{X}_{k+1} - \vec{X}_k$ then the following equation will be applied:

$$\vec{\partial} f_{k+1} = \vec{\partial} f_k + \vec{\partial}\vec{\partial}^{\mathrm{T}} f_k \cdot (\vec{X}_{k+1} - \vec{X}_k) + \mathcal{O}(\vec{X}_{k+1} - \vec{X}_k) \tag{1.1.7}$$

and rearranged:

$$\vec{\partial}\vec{\partial}^{\mathrm{T}} f_k \cdot (\vec{X}_{k+1} - \vec{X}_k) \approx \vec{\partial} f_{k+1} - \vec{\partial} f_k \; . \tag{1.1.8}$$

The approximate Hessian matrix must fulfill this condition, which is also known as the secant equation using the following transformation:

$$\hat{B}_{k+1} \cdot \vec{s}_k = \vec{y}_k; \quad \text{with} \quad \vec{s}_k = \vec{X}_{k+1} - \vec{X}_k; \quad \text{and} \quad \vec{y}_k = \vec{\partial} f_{k+1} - \vec{\partial} f_k \; . \tag{1.1.9}$$

The approximate Hessian matrix can be updated by a rank-one matrix at each iteration step k:

$$\hat{B}_{k+1} = \hat{B}_k + \gamma_k \, \vec{u}_k \cdot \vec{u}_k^{\,\mathrm{T}} \; . \tag{1.1.10}$$

Then, it follows that

$$\vec{u}_k = \vec{y}_k - \hat{B}_k \cdot \vec{s}_k; \qquad \text{and} \qquad \gamma_k = \frac{1}{\vec{u}^{\,\mathrm{T}} \cdot \vec{s}_k} \; , \tag{1.1.11}$$

and $\hat{B}_k$ is updated at each iteration step k as

$$\hat{B}_{k+1} = \hat{B}_k + \frac{(\vec{y}_k - \hat{B}_k \cdot \vec{s}_k)(\vec{y}_k - \hat{B}_k \cdot \vec{s}_k)^{\mathrm{T}}}{(\vec{y}_k - \hat{B}_k \cdot \vec{s}_k)^{\,\mathrm{T}} \cdot \vec{s}_k} \; . \tag{1.1.12}$$

However, this approach has the fundamental disadvantage that the Hessian matrix is not necessarily positive definite and that the denominator can become zero. Therefore, mostly a rank-two matrix is used for the updates.

1.1.5 BFGS Algorithm

A rank-two approach is given by the BFGS algorithm. This algorithm is named after Broyden [9], Fletcher [10], Goldfarb[11], and Shanno [12]. In this method, the Hessian matrix is updated with the following attempt:

$$\hat{B}_{k+1} = \hat{B}_k + \gamma_k \vec{u}_k \cdot \vec{u}_k^{\,\mathrm{T}} + \delta \vec{v}_k \cdot \vec{v}_k^{\,\mathrm{T}} \; ; \tag{1.1.13}$$

$$\vec{u}_k = \vec{y}_k; \quad \vec{v}_k = \hat{B}_k \cdot \vec{s}_k; \quad \gamma_k = \frac{1}{\vec{y}_k^{\,\mathrm{T}} \cdot \vec{s}_k}; \quad \delta_k = \frac{-1}{\vec{s}_k^{\,\mathrm{T}} \cdot \hat{B}_k \cdot \vec{s}_k} \; . \tag{1.1.14}$$

Even if the two update matrices in Eq. (1.1.13) are only rank-one, the sum is a rank-two matrix. Finally, the following update formula is derived:

$$\hat{B}_{k+1} = \hat{B}_k + \frac{\vec{y}_k \cdot \vec{y}_k^{\,\mathrm{T}}}{\vec{s}_k^{\,\mathrm{T}} \cdot \vec{y}_k} - \frac{\hat{B}_k \cdot \vec{s}_k \cdot \vec{s}_k^{\,\mathrm{T}} \cdot \hat{B}_k}{\vec{s}_k^{\,\mathrm{T}} \cdot \hat{B}_k \cdot \vec{s}_k} \; . \tag{1.1.15}$$

The BFGS algorithm can now be formulated: The initial setting is done by starting at an arbitrary or predefined point $\vec{X}_0$, and using the starting Hessian matrix $\hat{B}_0 = \mathbb{1}_{D\times D}$. Because $\hat{B}_0$ is the unit matrix, the first step is simply done in the direction of a steepest-descent step. However, with each iteration step, the Hessian matrix is filled with information about the function f. So the history of all already computed steps is used in addition to the simple gradient for the suggestion of the next search direction.

The following steps are repeated until the desired accuracy is reached:

- Suggest a new direction $\vec{p}_k$ by solving the equation system $\hat{B}_k \cdot \vec{p}_k = -\vec{\partial} f(\vec{X}_k)$.
- Determine the step length A_k, *e.g.*, through a line search: $\vec{X}_{k+1} = \vec{X}_k + A_k \vec{p}_k$.
- Determine the values of $\vec{s}_k = A_k \vec{p}_k$ and $\vec{y}_k = \vec{\partial} f(\vec{X}_{k+1}) - \vec{\partial} f(\vec{X}_k)$.
- Update the approximated Hessian matrix $\hat{B}_k$ according to Eq. (1.1.15).

For each iteration step, the algorithm only needs the current position $\vec{X}_k$ and the gradient $\vec{g}_k = \vec{\partial} f(\vec{X}_k)$ to provide a new position $\vec{X}_{k+1}$ which is probably closer to the minimum.

The line search is extremely important for the first steps, since no information about previous steps is available. Many applications omit the line search and use exactly $A = 1$ along the search direction. This is usually possible when the search vector $\vec{p}_k$ provides information on the length of the steps like in the BFGS-algorithm. At the beginning of the calculation, however, this specification is not given by the still inaccurate Hessian matrix.

The advantage of neglecting the line search is that every function call is used to improve the approximated Hessian matrix. However, this includes also the possibility to move away from the minimum. Normally this behavior is not problematic but for the conversion, the BFGS Algorithm requires a well defined function f in the considered space.

In this work, the function (as described in detail in section 2.3) is not free of definition gaps. By choosing a too large step length A, the $\vec{X}_{k+1}$ point at the next iteration could lay in such an undefined definition gap. To prevent this undesired behavior, here, a line search is always used instead of $A = 1$ for the initial iterations.

1.2 Total-Energy Calculations

The application of the optimization algorithms presented in the previous sections to a specific material, *e.g.*, a crystal, requires the definition of the function to be optimized. In this work, this function is identified with the total energy E_{TOT} of the crystal. This and the next sections are dedicated to the finding of an expression of the total energy and to the description of the theoretical methods which allows for its calculation.

From the theoretical point of view, a crystal can be seen as a many-body system of electrons and atomic nuclei interacting to each other through the Coulomb potential. Due to the presence of the electrons, the crystal has to be treated quantum

mechanically. In quantum mechanics a solid is defined by the following Schrödinger eigenvalue problem:

$$\hat{H}\,\Psi(\vec{r},\vec{R}) = E_{\mathrm{TOT}}\,\Psi(\vec{r},\vec{R})\,. \tag{1.2.1}$$

Here, $\vec{r} = \{\mathbf{r}_i; i = 1, 2, \ldots, N\}$ and $\vec{R} = \{\mathbf{R}_I; I = 1, 2, \ldots, N_{\mathrm{nuc}}\}$ are sets of all electron and nuclear coordinates, respectively. The total Hamiltonian $\hat{H}$ is defined as:

$$\hat{H} = \hat{T}_{\mathrm{nuc}} + \hat{V}_{\text{nuc-nuc}} + \hat{H}_e\,, \tag{1.2.2}$$

where, using overall atomic Hartree units ($\hbar = m_e = e = 1/4\pi\epsilon_0 = 1$) :

$$\hat{H}_e = \hat{T}_e + \hat{V}_{e\text{-}e} + \hat{V}_{e\text{-nuc}}\,, \tag{1.2.3}$$

$$\begin{aligned}
\hat{T}_{\mathrm{nuc}} &= -\sum_{I=1}^{N_{\mathrm{nuc}}} \frac{\boldsymbol{\nabla}_I^2}{2M_I}\,,\\
\hat{T}_e &= -\sum_{i=1}^{N} \frac{\vec{\nabla}_i^2}{2}\,,\\
\hat{V}_{\text{nuc-nuc}} &= \frac{1}{2}\sum_{I=1}^{N_{\mathrm{nuc}}}\sum_{\substack{J=1\\I\neq J}}^{N_{\mathrm{nuc}}} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|}\,,\\
\hat{V}_{e\text{-}e} &= \frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\i\neq j}}^{N} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}\,,\\
\hat{V}_{e\text{-nuc}} &= -\sum_{i=1}^{N}\sum_{I=1}^{N_{\mathrm{nuc}}} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \equiv \sum_{i=1}^{N} v(\mathbf{r}_i)\,.
\end{aligned} \tag{1.2.4}$$

In the equations above, Z_I and Z_J are the charge and M_I is the mass of the corresponding nucleus I or J.

Typically, in a solid of a volume of 1 cm^3, the number of atoms is in the order of magnitude of 10^{23}. Therefore, in order to reduce the complexity of the problem, some further approximations are needed. A first reduction of the degrees of freedoms is obtained by the Born-Oppenheimer approximation. Due to the fact that electrons have much smaller mass and much higher velocities compared to the nuclei, it can be assumed that the motion of atomic nuclei can be decoupled from the motion of electrons. Within this approximation, the wave-function of the total system can be separated into a part corresponding to the atomic nuclei and a part describing the electrons

$$\Psi_{\mathrm{BO}}(\vec{R},\vec{r}) = \chi_{\mathrm{nuc}}(\vec{R})\,\phi_e(\vec{r},\vec{R})\,. \tag{1.2.5}$$

The electronic wave-function $\phi_e(\vec{r},\vec{R})$ is the eigenfunction of the electronic Hamiltonian $\hat{H}_e$

$$\hat{H}_e\,\phi_e(\vec{r},\vec{R}) = E_e(\vec{R})\,\phi_e(\vec{r},\vec{R})\,, \tag{1.2.6}$$

where E_e is the electronic energy. The explicit, parametric dependence of the electronic wave-functions ϕ_e on the fixed nuclear configuration $\vec{R}$ will be omitted in the following by considering the influence of the nuclei as external potential.

Assuming the electronic system to be in the ground state corresponding to the nuclear configuration $\vec{R}$, the nuclear wave-function $\chi_{\text{nuc}}(\vec{R})$ satisfies the Schröedinger equation

$$\left[\hat{T}_{\text{nuc}} + \hat{V}_{\text{nuc-nuc}}(\vec{R}) + E_{e,\text{gs}}(\vec{R})\right] \chi_{\text{nuc}}(\vec{R}) = E_{\text{BO}}\, \chi_{\text{nuc}}(\vec{R})\,. \tag{1.2.7}$$

For a fixed nuclear configuration $\vec{R}$, according to this eigenvalue equation, a corresponding "total" potential energy can be defined as

$$E_{\text{TOT}}(\vec{R}) = \hat{V}_{\text{nuc-nuc}}(\vec{R}) + E_{e,\text{gs}}(\vec{R})\,. \tag{1.2.8}$$

The configuration $\vec{R}$ which minimizes this energy is the optimal configuration which will be determined. The explicit calculation of the total energy $E_{\text{TOT}}(\vec{R})$ is done by using the density-functional theory which is introduced in the next section.

1.2.1 Density-Functional Theory

According to Eq. (1.2.8), the explicit ground state of the system $E_{e,\text{gs}}(\vec{R})$ is required. The Hohenberg-Kohn theorem [13, 14] shows that the ground-state energy of a system is a functional of the electron density n. It shows also that the ground-state energy becomes minimal only with the ground-state electron density n_{gs}

$$E_{\text{gs}}[n] \geq E_{\text{gs}}[n_{\text{gs}}]\,. \tag{1.2.9}$$

The electronic energy is given according to the Hamilton operator in Eq. (1.2.3) and the eigenvalue problem in Eq. (1.2.6) by the following functional components:

$$E_{e,\text{gs}}[n] = T_e[n] + V_{e\text{-}e}[n] + E_{\text{ext}}[n]\,. \tag{1.2.10}$$

The external energy functional E_{ext} refers to the external potential $v(\mathbf{r})$

$$E_{\text{ext}}[n] = \int n(\mathbf{r})\, v(\mathbf{r})\, d^3r\,, \tag{1.2.11}$$

which includes the potential of the nuclei to the electrons $V_{e\text{-nuc}}$ and, in principle it can contain external fields, too.

Since the exact kinetic functional is unknown, it is approximated as

$$T_e[n] \equiv T_0[n] + \alpha[n]\,, \tag{1.2.12}$$

with T_0 as the kinetic-energy functional of a "non-interacting" electron system with the density $n(\mathbf{r})$ and the unknown part of the interaction is saved in $\alpha[n]$. For the electron-electron interaction, an Hartree contribution E_{H} is introduced:

$$V_{e\text{-}e}[n] = \frac{1}{2}\int \frac{n(\mathbf{r}_1)\, n(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 + \beta[n] \equiv E_{\mathrm{H}}[n] + \beta[n] \; . \tag{1.2.13}$$

So the energy functional can be rewritten as follows:

$$E_{e,\mathrm{GS}}[n] = T_0[n_{\mathrm{gs}}] + E_{\mathrm{H}}[n_{\mathrm{gs}}] + E_{\mathrm{xc}}[n_{\mathrm{gs}}] + E_{\mathrm{ext}}[n_{\mathrm{gs}}] \; . \tag{1.2.14}$$

The kinetic term is $T_0[n_{\mathrm{gs}}]$ and the Hartree energy is $E_H[n_{\mathrm{gs}}]$. The energy from external sources $E_{\mathrm{ext}}[n]$ remains unchanged and the error of the Hartree energy and kinetic energy becomes the exchange-correlation functional:

$$E_{\mathrm{xc}}[n] = \alpha[n] + \beta[n] \; . \tag{1.2.15}$$

The ground-state energy is obtained according to the Hohenberg-Kohn theorem by minimizing the energy functional for any electron density $n(\mathbf{r})$ with the constraint

$$\int n(\mathbf{r})\, d^3r = N \; . \tag{1.2.16}$$

The ground-state energy is the lowest possible energy of the system

$$E_{\mathrm{gs}} = \min_{n(\mathbf{r})} E_{e,\mathrm{gs}}[n] \; . \tag{1.2.17}$$

To minimize the energy with respect to the electron density, the Euler-Lagrange formalism is used. Here, the auxiliary function is

$$A(\mu, [n]) = E_{\mathrm{gs}}[n] - \mu \left[\int n(\mathbf{r}) d^3r - N \right] \; , \tag{1.2.18}$$

$$\left. \frac{\delta E_{\mathrm{gs}}[n]}{\delta n(\mathbf{r})} \right|_{n_{\mathrm{gs}}} = \mu \; , \tag{1.2.19}$$

with μ as the Lagrange multiplier. If the energy terms from Eq. (1.2.14) are formally entered, the following formula will be derived:

$$\frac{\delta T_0[n]}{\delta n(\mathbf{r})} + \underbrace{\frac{\delta E_{\mathrm{H}}[n]}{\delta n(\mathbf{r})}}_{v_{\mathrm{H}}(\mathbf{r},[n])} + \underbrace{\frac{\delta E_{\mathrm{xc}}[n]}{\delta n(\mathbf{r})}}_{v_{\mathrm{xc}}(\mathbf{r},[n])} + \underbrace{\frac{\delta E_{\mathrm{ext}}[n]}{\delta n(\mathbf{r})}}_{v(\mathbf{r})} = \mu \; , \tag{1.2.20}$$

and, thus, the so-called Kohn-Sham potential (KS) [15]

$$v_{\mathrm{KS}}(\mathbf{r}, [n]) = v(\mathbf{r}) + v_{\mathrm{H}}(\mathbf{r}, [n]) + v_{\mathrm{xc}}(\mathbf{r}, [n]) \; . \tag{1.2.21}$$

The individual potential terms are now one-particle potentials. With this formulation, any interactions between the electrons are not made explicitly, but inserted in the new external potential v_{KS}. The Kohn-Sham potential indeed includes the effects of interactions

$$\frac{\delta T_0[n]}{\delta n(\mathbf{r})} + v_{\mathrm{KS}}(\mathbf{r}, [n]) = \mu \,. \tag{1.2.22}$$

Thus, with a given Kohn-Sham potential, the electronic energy can be minimized to obtain the ground-state electron density according to this potential. The electron density can be determined as

$$n(\mathbf{r}) = \sum_{j}^{N} |\psi_j^{\mathrm{KS}}(\mathbf{r})|^2 \,, \tag{1.2.23}$$

where the one-particle wave functions ψ_j^{KS} are obtained by the solution of the Kohn-Sham equations:

$$\left[-\frac{\boldsymbol{\nabla}^2}{2} + v_{\mathrm{KS}}(\mathbf{r})\right] \psi_j^{\mathrm{KS}}(\mathbf{r}) = \epsilon_j^{\mathrm{KS}}\, \psi_j^{\mathrm{KS}}(\mathbf{r}) \,. \tag{1.2.24}$$

The above single-particle equations are solved self-consistently using the scheme:

- Starting with an initial guess for the electron density of the system, the KS potential is determined by Eq. (1.2.21).
- By solving the eigenvalue problem in Eq. (1.2.24), the KS eigenstates ψ_j^{KS} are determined.
- The ground-state electron density $n(\mathbf{r})$ is computed from Eq. (1.2.23).
- Using the ground-state electron density a new KS potential can be determined.
- Solve again the KS equations and find a new $n(\mathbf{r})$.
- This is repeated until the ground-state electron density $n(\mathbf{r})$ has sufficiently converged.

Within the Born-Oppenheimer approximation, the mathematical model of the DFT is exact. However, there exist no explicit functional form for the exchange-correlation potential $v_{\mathrm{xc}}[n]$. Thus, this functional has to be approximated.

There are currently many approaches to do this. In the local-density approximation (LDA), the exchange-correlation energy depends on the local value of the electron density only. The semi-localized generalized gradient approximation (GGA) extends LDA by the gradient of the electron density. MetaGGA extends the GGA with the kinetic-energy density. Though relevant for specific applications, higher-order approximations than GGA are not considered in this work.

1.2.2 Local-Density and General Gradient Approximation

To use $v_{\mathrm{xc}} = 0$ as the lowest-order approximation yields very bad results. The first reasonable approximation for the exchange-correlation potential v_{xc} is the local-density approximation. Here, the exchange-correlation energy per particle is obtained by considering the result for the uniform electron gas. Thus, this approximation is exact for a homogeneous electron gas (h), and it is a good approximation for electron densities with weak changes. $E_{\mathrm{xc}}^{\mathrm{LDA}}$ is, then, a pure functional of the electron density:

$$E_{\mathrm{xc}}^{\mathrm{LDA}}[n] = \int n(\mathbf{r}) \left[\epsilon_{\mathrm{x}}^{h}(n(\mathbf{r})) + \epsilon_{\mathrm{c}}^{h}(n(\mathbf{r}))\right] d^3r \ . \tag{1.2.25}$$

The exchange part $\epsilon_{\mathrm{x}}^{h}(n)$ is an analytical function of the density:

$$\epsilon_{\mathrm{x}}(n) = -\frac{3}{4\pi}\left(3\pi^2 n\right)^{\frac{1}{3}} \ . \tag{1.2.26}$$

On the other side, there is no analytical expression for the correlation contribution except for the high- and low-density limits corresponding to the infinite-weak and infinite-strong correlation. For values in between the limits interpolations or Monte Carlo calculations must be made.

The next order of approximation is the general gradient approximation. This approximation is used to improve the LDA by including the gradients of the density to capture the non-homogenity of the real density.

$$E_{\mathrm{xc}}^{\mathrm{GGA}}[n] = \int \epsilon_{\mathrm{xc}}^{\mathrm{GGA}}\Big[n(\mathbf{r}), \boldsymbol{\nabla} n(\mathbf{r})\Big]\, n(\mathbf{r})\, d^3r \tag{1.2.27}$$

The function $\epsilon_{\mathrm{xc}}^{\mathrm{GGA}}$ depends on a set of parameters which are found by assuming that the $E_{\mathrm{xc}}^{\mathrm{GGA}}$ functional satisfies some general properties of the exact exchange-correlation energy functional. Among the several GGA functionals which have been proposed [16, 17, 18], in this work, the Perdew-Burke-Ernzerhof (PBE) functional [19] is used.

1.2.3 Van-der-Waals Interaction

In molecular crystals, two different kinds of interactions can be identified. The first ones are the short-range but strong binding interactions between the atoms of a given molecule. The second type of interaction is represented by the long-range but weak van-der-Waals forces between the atoms in different molecules.

Functionals like LDA or GGA do not describe the long-range electron correlations that are responsible for the van-der-Waal interaction. Even if the interaction

is very weak, it plays a central role in many chemical systems. This is the reason why it has to be taken into account.

In this thesis, the "`DFT-D2`" approach [20] is used. The energy E_{TOT} is corrected by an empirical energy term:

$$E_{\mathrm{TOT+VDW}} = E_{\mathrm{TOT}} + E_{\mathrm{VDW}} \ . \tag{1.2.28}$$

The correction term includes the long-range van-der-Waals interaction between the atoms I and J:

$$E_{\mathrm{VDW}} = -\frac{1}{2}\, s_6 \sum_{I \neq J} \frac{C_6^{IJ}}{(R_{IJ})^6}\, f_{\mathrm{DMP}}(R_{IJ}) \ . \tag{1.2.29}$$

Here, $C_6^{IJ} = \sqrt{C_6^I C_6^J}$ are the dispersion coefficients, which depend on the atom type of atom I and J and are listed in Ref. [20]. R_{IJ} is the distance between the two atoms I and J. s_6 is a global scaling factor for the dispersion coefficients. It depends on the exchange-correlation functional as used in DFT. f_{DMP} is a damping function to avoid near-singularities for small R_{IJ}.

A second approach is also used in this work. It is called "`TS-vdW`" [21]. In this case, the dispersion coefficients C_6^{IJ} are calculated using explicitly the electron density.

1.2.4 (Linear) Augmented Plane Waves and Local Orbitals

For an optimal numerical solution of the Kohn-Sham equations, the introduction of a problem-adapted set of basis functions is required. A first possibility is to use plane waves because of the periodicity of the crystal system. However, in the region close to the atoms, the single-electron KS wave functions orbital are strongly oscillating function. Then, for an accurate calculation a very large set of plane waves would be required. But in fact, even a huge basis would not converge the KS equations.

In order to avoid very large basis sets, the unit cell of the crystal can be separated into two spatial regions [22]. Around each atom, a sphere is introduced with the center at the atom position. Inside the spheres, in the so-called muffin-tin (MT) region, the KS wave functions are described by "atomic-like" basis functions, while the expansion in terms of plane waves is still used in the remaining region, the interstitial region (IR). A visualization of this separation can be found in Figure 2.3.1. The two kinds of basis functions are then imposed to be continuous on the MT spheres. This unique type of basis functions defined on the full unit cell of the crystal is called augmented plane wave (APW). The way in which the atomic-like function in the MT region is chosen, gives rise to many possibilities, among which, is worth to mention the linear APW (LAPW). In this case a combination of a solution

of the radial Schrödinger equation for the atom and its derivatives with respect to the energy is chosen.

In addition, to the (L)APWs, the quality of the basis set can be improved by adding a set of local orbitals (LO). The LO are only defined inside the MT region and vanish in the IR.

1.2.5 The exciting Code

All force and energy calculations were performed using the open-source full-potential all-electron code **exciting** [22]. This code implements the solution of the self-consistent Kohn–Sham equations. In addition, the concept of the dual (L)APW basis as described in Section 1.2.4 is used in the code. Further details can be found on the code web page [23].

Concerning the structural optimization, there are several methods already implemented in the **exciting** code. In the "`newton`" method, the atoms are displaced according to their gradients. In the algorithm "`harmonic`", a separation of the D degrees of freedom is performed. For each of them a parabolic energy curve is assumed. Finally, there is the "`LBFGS`" method, which is much the same as the BFGS method, except that the Hessian is no longer stored but assembled from the last adjustable number of calculation steps. The L in this name stands for low memory, because less memory is used with this method for not storing the whole Hessian but some step vectors $\vec{X}_k$.

1.3 Choise of the Test System: Anthracene

As a test molecule the for calculations, anthracene is chosen. The structure of the anthracene molecular crystal is well known and often studied in the past [24, 25, 26, 27]. Therefore, the results of the optimization methods present in this work can be easily compared to several experimental data as well as to previous calculations.

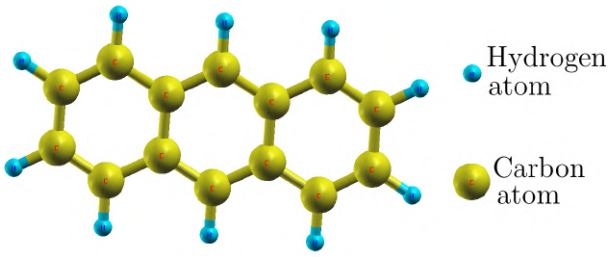

Figure 1.3.1: The structure of anthracene, an aromatic molecule. It consist of 10 hydrogen and 14 carbon atoms.

Anthracene is a molecule as shown in Figure 1.3.1 consisting of three hydrocarbon rings. Adjacent rings share two carbon atoms. All the atoms of this molecule are arranged in a plane. When anthracene crystallizes, it forms a monoclinic structure with space group $P2_1/a$ as shown in Figure 1.3.2.

The experimentally determined structure has two molecules in the monoclinic unit cell with lattice parameters $a = 8.562\,\text{Å}$, $b = 6.038\,\text{Å}$, $c = 11.184\,\text{Å}$, and $\beta = 124.7°$ [25]. The orientation of the molecules in the unit cell are defined by the three angles θ, δ, and χ. These angles change when pressure is applied to the crystal. The angle convention from Ref. [24] as described in Figure 1.3.3 is used.

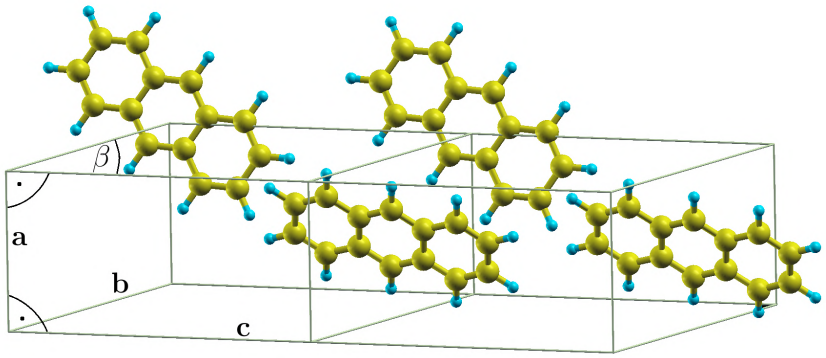

Figure 1.3.2: Structure of molecular crystal anthracene in its monoclinic unit cell.

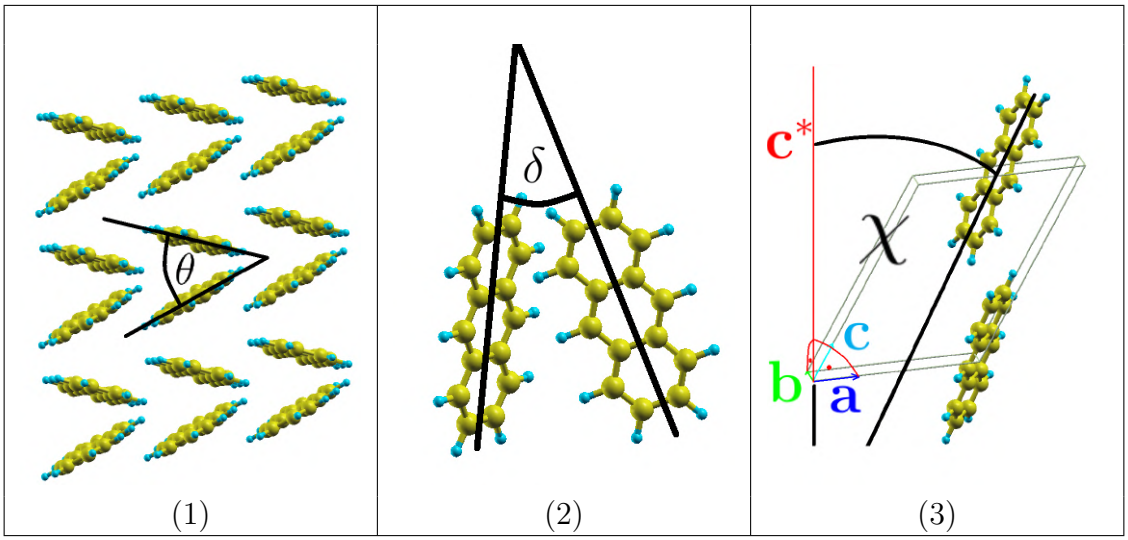

Figure 1.3.3: The structure of anthracene molecules and the angle definition as described in [24]. (1) The herringbone angle θ is the intersection angle between the planes describing the two translationally inequivalent molecules from a unit cell. (2) δ is the intersection angle of the two long molecule axes of the two translationally inequivalent molecules. (3) χ is the angle from the long molecule axis to the $\mathbf{c}^*$ axis, which is orthogonal to the plane defined by the basis vectors $\mathbf{a}$ and $\mathbf{b}$.

Chapter 2

Implementation

For this work, a new full BFGS method has been created. This new method, firstly, works exact like the already implemented method LBFGS. Then, the reduction of the degrees of freedom for rigid molecules is introduced and finally the complete description of the full molecular degrees of freedom is discussed. In addition, an algorithm to identify the molecules from a cloud of points will be described shortly.

2.1 BFGS

In order to do the structure optimizations, the BFGS algorithm is used as a kind of an exchangeable black box. The function value $f(\vec{X}_k)$ to be minimized and its gradient $\vec{g}_k = \vec{\partial} f(\vec{X}_k)$ are required at each optimization step k. This information is passed to the BFGS routine. This returns the next suggested position $\vec{X}_{k+1}$. The function to be minimized in the crystal is the total energy $f = E$. The computed energy depends on almost everything that is supplied to the **exciting** program input. In theory all **exciting** variables like "`ngrik`", "`rgkmax`" or "`scale`" can be used as degrees of freedom to minimize the energy. This means that these variables combined in a high-dimensional space spans the energy function whose degrees of freedom are varied in order to find the energy minimum.

In practice, however, the calculation requires the gradients of the energy according to these degrees of freedom and that turns out to be much more difficult. This work is limited to the structure optimization, which is significantly determined by the space position of the atoms in the crystal lattice. As free degrees of freedom, the $3N_{\text{Atom}}$ coordinate variables of N_{Atom} atoms in a lattice cell are used. Each atom i provide the degrees of freedom x_i, y_i and z_i to the structure optimization. Other dependencies such as lattice vectors are not explicitly specified here because they remain constant for a given relaxation. So the energy depends on them but these

derivatives are zero. Therefore, these variables will be neglected in the following work to focus on the important degrees of freedoms used for relaxation.

So the optimization vector $\vec{X}$ will be defined as follow:

$$\vec{X} = (\mathbf{r}_1^{\mathrm{T}}, \mathbf{r}_2^{\mathrm{T}}, \cdots, \mathbf{r}_{N_{\mathrm{Atom}}}^{\mathrm{T}})^{\mathrm{T}} \equiv \{x_i, y_i, z_i\,;\ i = 1, 2, \cdots, N_{\mathrm{Atom}}\}^{\mathrm{T}}\,, \tag{2.1.1}$$

a vector with $3N_{\mathrm{Atom}}$ single entries. The order of the entries does not matter but has to be constant for a complete relaxation. Therefore, the second notation of the vector $\vec{X}$ focused on the important components of the whole vector. So the degrees of freedom are splitted up in components where the gradients have similar behavior.

For the described algorithm here, however, the optimization vector is only a list of parameters. Moreover, the user can set certain single degrees of freedom to a fixed value. Atoms kept in a fixed position are marked via "`lockxyz`". The energy function to be minimized then no longer depends on these atomic positions anymore. This is, *e.g.*, useful if the user wants to prevent a drift of the structure in space while relaxing. In this way, the optimization vector $\vec{X}$ can looses entries since constant degrees of freedoms have always a zero derivative and must not be considered. Any reduction of the degrees of freedoms reduces the space of $f(\vec{X})$ for finding the desired local minimum and results in a reduced computation time. However, the desired minimum may not lay in the considered subspace of $f(\vec{X})$ anymore. Nevertheless, for a complete description, the full $3N_{\mathrm{Atom}}$ problem is assumed here.

The crystal energy is then defined by:

$$f(\vec{X}) = E(\vec{X}) = E(\{x_i, y_i, z_i\,;\ i = 1, 2, \cdots, N_{\mathrm{Atom}}\}^{\mathrm{T}})\,. \tag{2.1.2}$$

The calculation of the energy $E(\vec{X})$ is performed using the code **`exciting`**. This represents the most time-consuming part of the optimization procedure. The starting values of the degrees of freedom are the start position coordinates:

$$\mathbf{r}_i^{k=0} = (x_i^{k=0}, y_i^{k=0}, z_i^{k=0})^{\mathrm{T}}\,, \tag{2.1.3}$$

as specified in the input file for every atom i. The gradient of the energy is according to the $3N_{\mathrm{Atom}}$ dimension vector $\vec{X}$ also a vector of the dimensionality of $3N_{\mathrm{Atom}}$:

$$\begin{aligned}
\vec{g}(\vec{X}) = \vec{\partial} E(\vec{X}) &= \vec{\partial} E(\{x_i, y_i, z_i\,;\ i = 1, 2, \cdots, N_{\mathrm{Atom}}\}^{\mathrm{T}}) \\
&= \left\{ \frac{\partial E}{\partial x_i}, \frac{\partial E}{\partial y_i}, \frac{\partial E}{\partial z_i}\,;\ i = 1, 2, \cdots, N_{\mathrm{Atom}} \right\}^{\mathrm{T}} \\
&= -\left\{ F_{x_i}, F_{y_i}, F_{z_i}\,;\ i = 1, 2, \cdots, N_{\mathrm{Atom}} \right\}^{\mathrm{T}} \quad \text{, because}
\end{aligned} \tag{2.1.4}$$

$$\frac{\partial E}{\partial x_i} = -F_{x_i} \quad \frac{\partial E}{\partial y_i} = -F_{y_i} \quad \frac{\partial E}{\partial z_i} = -F_{z_i} \text{ , and thus}$$
$$\frac{\partial E}{\partial \mathbf{r}_i} = -\mathbf{F}_i \ . \tag{2.1.5}$$

The gradients of the energy are proportional to the force $\mathbf{F}_i$ acting on a single atom i. The force components on the atoms are included in an energy calculation in **exciting** and can be used here.

To summarize, the following optimization loop is used in this work:

- First, the atomic coordinates are collected as degrees of freedom in the optimization vector $\vec{X}$. At least, one degree of freedom must be present.
- Then, the initial values $E(\vec{X}_{k=0})$ and $\vec{\partial}E(\vec{X}_{k=0})$ are passed to the optimization routine.
- The optimization routine returns a new optimization vector $\vec{X}_{k+1}$ which delivers a new atomic configuration.
- For the new configuration, a new $E(\vec{X}_k)$ and $\vec{\partial}E(\vec{X}_k)$ are calculated and passed to the routine.
- This optimization loop will be repeated until a specified accuracy is reached, *e.g.*, if the absolute value of the maximum force components is below a desired value.

If the algorithm converges, then by definition of a minimum, the gradient should go to zero. In the **exciting** code, the default value for the maximum force components to be considered as zero is $2 \cdot 10^{-4}$ Ha/Bohr.

2.2 Identification of Molecules

In the case of molecular crystals, it is important to identify to which molecule an atom from the unit cell corresponds. To do this, a simple cluster-formation algorithm is used here. The algorithm uses a search length as an input parameter. It starts by assuming that each single atom is its own small cluster. Then, a sphere around each atom with the radius of this input parameter is drawn. All other atoms within this sphere are now added to the cluster of the considered atom. This is repeated for all atoms until there is no change in the clusters affiliation. Thus, all start-up clusters are condensed iteratively until its atoms have no neighbors of other clusters left within the defined radius.

In order to obtain the definition of molecules in terms of clusters with this algorithm, the input parameter must satisfy two important conditions.

First, the search length must be greater than the bond length between the atoms. Otherwise, the algorithm ends up right after the start and returns all the atoms in its own cluster, which does not represent any gain of information. Second, the search length must not be greater than the smallest distance between the molecules. Otherwise, the algorithm iterates until all the atoms are in the same cluster, which also does not represent any gain of information.

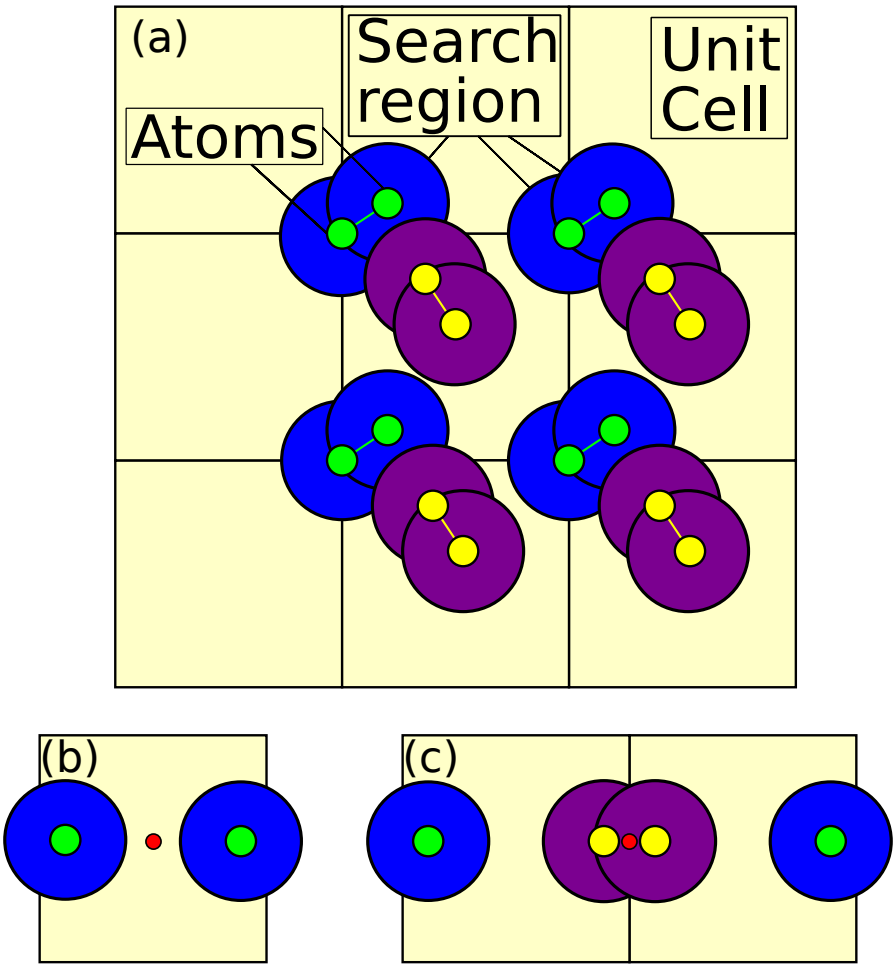

Figure 2.2.1: Visualization of the clustering algorithm. (a) All atoms that are closer than the search parameter to another atom belong to the same cluster. This also applies beyond the limits of the unit cell. Moreover, the molecule detection is correct in the repeated cell representation only. (b) The position of the molecule center (red circle) can be wrong for a molecule which atoms are only in the unit cell. (c) If the atoms out of the unit cell are involved, the center of the molecule can be calculated correctly.

Oligo-acenes molecules like anthracene can be reliably identified with this method, but the quality of the algorithm depends strong on the parameter of the maximum search distance entered manually.

Determining this parameter automatically is difficult because there is no criterion for an optimum number of clusters. The number of clusters to be used ranges

between the limits of one cluster of all atoms of its own cluster. If the result does not satisfactorily identify the molecules, the input parameter has to be changed until the molecules can be correctly identified.

With anthracene, this algorithm is definitely possible to use but other crystals may have problems. The problems occur, *e.g.*, in the case that the bond length between some atoms of a molecule is longer than the distance to another molecule. For these cases the algorithm can never deliver the correct results regardless of the search parameter. This clustering algorithm is therefore applicable to the problems investigated in this work but is not universally usable. In general, the user must manually make sure that the membership of the atoms to the molecules is correct.

At start of the algorithm the unit cell must be replicated in all spatial directions for the neighborhood analysis. This is done because a molecule does not have to stick to the edges of the unit cell. In this work the given atomic position is almost always outside the unit cell. Thus, the replication must include several unit cells to compare all positions of the atoms to each other properly.

For the determination of the molecular center of mass it is important to allow the atomic position to be outside the unit cell. The center of mass $\mathbf{M}_m$ of a molecule depends on the exact position $\mathbf{r}_i$ of the atoms i within the molecule m and the atomic mass m_i:

$$\mathbf{M}_m = \frac{\sum_{i \in m} m_i \mathbf{r}_i}{\sum_{i \in m} m_i} \,. \tag{2.2.1}$$

For every atom outside the unit cell, there is a lattice translation that can translate the atom within the unit cell without changing the overall structure. But the atom on this position can belong to another molecule. In this case, the center of mass calculation will be wrong due to the fact that wrong atom positions will be considered.

Suppose that a molecule center lies exactly in the middle on the edge of the unit cell. If the atoms were positioned correctly within the unit cell and the center of mass of the molecule is calculated from this information, then this point would be in the middle of the unit cell because the molecule is equally present on the right and on the left. But this is wrong because the center of the molecule is per definition in the middle on the edge of the unit cell. This fact is visualized in figure 2.2.1 (b) and (c).

Since the program does not know, which position of the molecular atom is the correct position for the molecule, the position have to be correct by input, even if the position is not arranged within the unit cell. Therefore, the ability for atoms to be outside of the unit cell is required. This works well for the program **`exciting`**, if the variable "`tshift`" is set to false AND carthesian coordinates are used. If this

requirement is not present, then **exciting** donates the atoms back to the unit cell. The result is a wrong center of mass calculation for all splitted molecules.

2.3 Input conditions

As described in 1.2.4, the **exciting** program essentially splits the unit cell into the muffin-tin regions with atomic orbital basis and the interstitial region in the basis of plane waves.

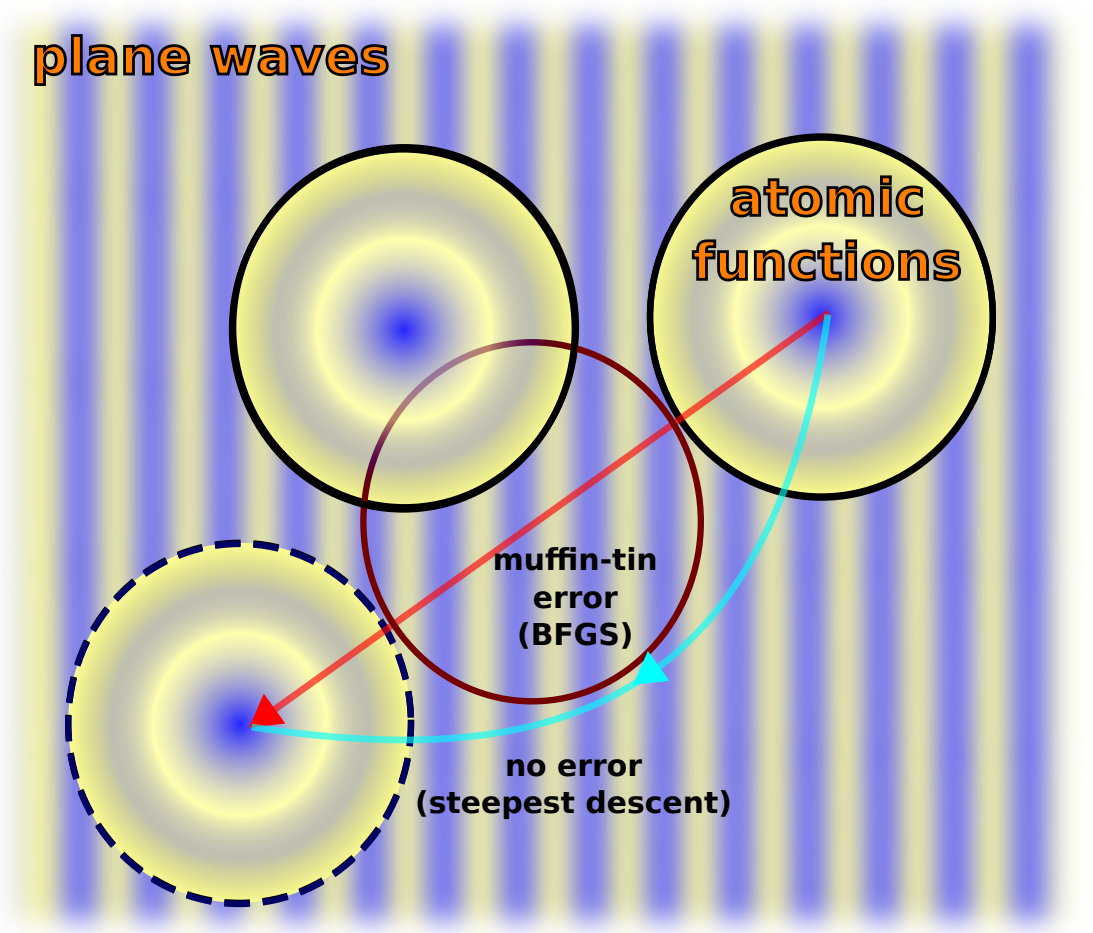

Figure 2.3.1: Visualization of the different regions of the crystal simplified to 2D. Within the radius, the wave function is approximated from orbital basis functions and outside with planar wave functions. The energy can only be calculated if the muffin-tins do not touch. For the BFGS method, those are the areas where the $f = E$ function is undefined, which can cause problems. Since the undefined places only occur when two atoms come very close and thus the energy is very large according to the Lenard-Jones potential, this area is theoretically not relevant for an energy minimization. Simple gradient methods should not be able to reach these areas, but the BFGS method can thus hit a definition gap.

In order to know the exact basis in a region, the muffin-tins of different atoms must not overlap. Otherwise, there exist two definitions of a basis in the overlapping

region and this is not allowed. If two atoms come too close, the radius of the muffin-tins must be made smaller to avoid the overdetermined space. For unit-cell optimization, there are areas where the energy function is undefined. This occur exactly where two muffin-tin radii would overlap. This is called muffin-tin error.

However, the BFGS method requires for successful application, that the function to be minimized is well defined in the whole space considered. This is not the case if two atoms come to close. Therefore, it must be sure that the positions proposed by the BFGS method must not reached the undefined areas. During the relaxation, the atoms strive for an equilibrium position. When these atoms get too close, a huge potential barrier appears. If the atomic radii are suitable selected, the undefined area should physically not be reached.

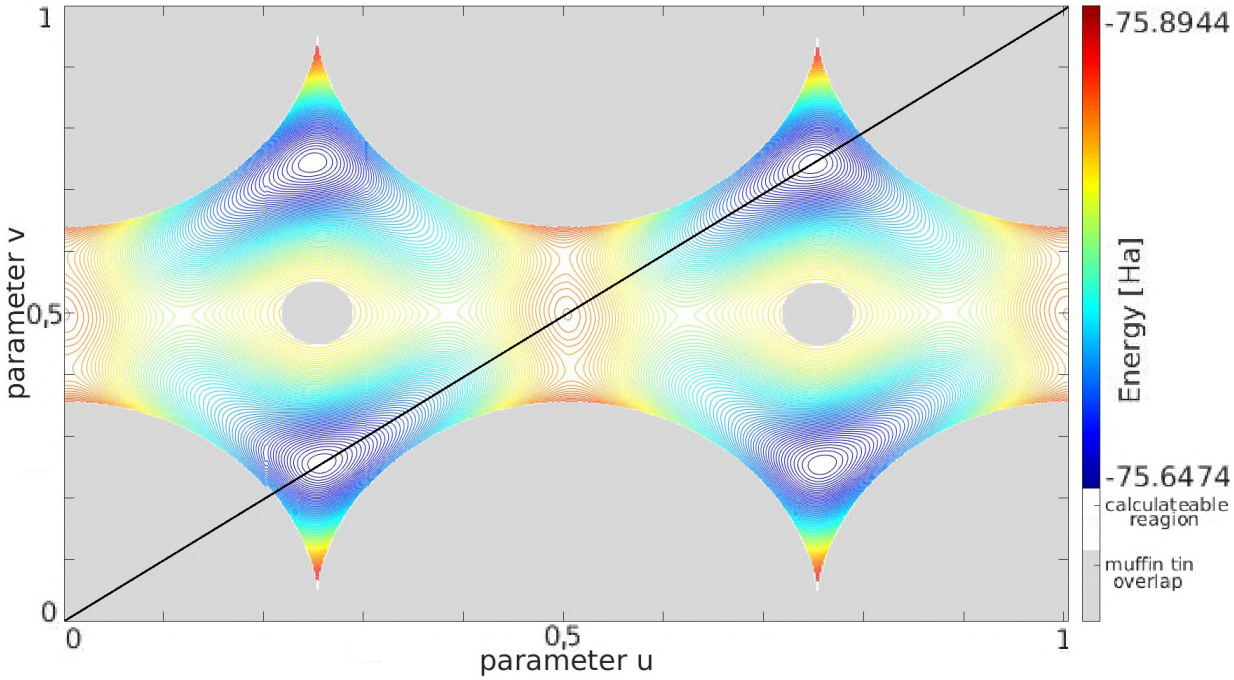

Figure 2.3.2: The parameterized energy surface of the bulk diamond structure. One carbon atom was fixed in the $(0,0,0)$ point of the face-centered cubic Bravais lattice. The other carbon atom was moved along the parametrization $\mathbf{r}_c = (0,0,0)^{\mathrm{T}} + u(1,0,0)^{\mathrm{T}} + v(0,1,1)^{\mathrm{T}}$ in the units of the lattice vectors and the energy of each system was measured. This parameterized plane through the unit cell contains the diagonal line (black line in the figure) of the unit cell and so the expected energy minimum at a quarter of this line. For visualization, the parameterized surface is only a 2 dimensionality space rather than the three possible degrees of freedom for moving one atom. The gray regions are undefined regions because the two atoms come too close to each other. The BFGS algorithm has not to propose the next search point in these regions. Especially here, the undefined region are close to the expected minima in blue.

However, there are several scenarios possible where two atoms may come too close.

- If a too large step size A has been selected by the BFGS line search, then two oncoming atomic radii may overlap.
- If, as in Figure 2.3.1, the directional proposal from the BFGS routine passes an atom, the path to this atom may result in a muffin-tin error.
- If the muffin-tin radius is chosen too large, the atoms of a relaxed structure must overlap, results in a muffin-tin error also.

In general, it is always a good choice to place the starting positions of the atoms close to the expected energy minimum. The BFGS algorithm converges at near-square convergence if the minimum is close. Here no muffin-tin error is expected.

However, if there is no knowledge about an expectation of the energy minimum, then the BFGS algorithm uses many function calls. The algorithm scans the local energy surface only and needs a long way to find the minimum. Mathematicians call such a configuration ill-conditioned [28]. Moreover, using such an input, it is impossible to predict in which local minimum the system will end up. If the algorithm finds a minimum and does not get stuck in definition problems, it is not sure to end up in a desired minimum. *E.g.*, for crystals with different phases such as Carbon in diamond structure or graphite the result is unpredictable for an so called ill-conditioned input. Only a local minimum can be found. Ill-conditioned inputs require exponentially longer computational time because the rate of convergence drops to extremely low values. So a lot of time can be saved by giving the algorithm the minimum it should converge to as accurately as possible.

2.4 Gradient Calculation

Supposed the function f to minimize is not exactly known. Then, the function values can not be calculated directly. However, with the DFT, there exist a tool to determine the function $f^{\text{approx}} \approx f = E$ as an approximated value. The function values depend extremely on the discretization and the approximation of the problem [29, 30]. The number of basis functions to be used for the calculation ("rgkmax") or also the value of grid density of the calculation ("ngridk") are examples for controlling the discretization. The used density functionals are examples for the approximations. The approximated function f^{approx} is the only given information. Therefore this function can be optimized only. But even this approximated function is not known analytically. Only a function value at a given position $\vec{X}$ within a ground state calculation can be determined. This value is an approximation of the unknown function $f(\vec{X})$.

A major factor for the minimization, however, is the quality of the gradient. In

general there are two different ways to calculate the gradient.

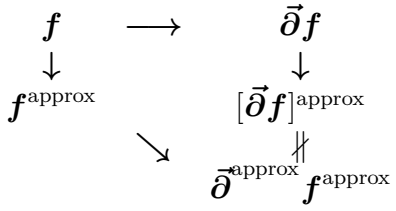

Figure 2.4.1: An overview of the unknown function f and the approximated functions f^{approx} which can be calculated. Both gradients are approximations to the gradient of the unknown function f. Depending on the way the approximation is done the gradients can be different.

First the gradient could be determined analytically from the approximated function f^{approx}. If this first way is possible, this is the best gradient that can be obtained since this gradient fits exactly to the function to be minimized. However, this is not always possible when, *e.g.*, the approximated function f^{approx} is not known.

But usually the internal differential equations can be solved formally to determine the gradient from the unknown function f. This gradient $\vec{\partial} f$ is also generally unknown and it is determined analog to the approximation and discretization of the main function f. Both gradients are approximations of the gradients of the unknown function f. But depending on the chosen approximation and the discretization, the gradient $[\vec{\partial} f]^{\text{approx}}$ thus obtained may deviate from the gradient $\vec{\partial}^{\text{approx}} f^{\text{approx}}$ of the function f^{approx} to be minimized. In the ideal case the two gradients are equal. In this case the approximation and discretization have no influence on the function values.

If the approximated gradient is too inaccurate, the vector does not point the direction to the minimum. However, in this work the approximated gradient $[\vec{\partial} f]^{\text{approx}}$ is used as gradient since the derivation of the unknown function is not possible. It must be sure that the approximation and discretization has a sufficiently small influence on the gradients. The calculation of the energy and the forces between the atoms must be sufficiently accurate. This is checked by energy convergence tests.

2.5 Anthracene Relaxation

To start a calculation, often experimental values, *e.g.*, x-ray scattering experiments may be used to get the first positions of the atoms in the unit cell. For Anthracene the unit cell and the starting positions are set up as in section 1.3 described. These starting positions of the atoms should be close enough to the desired minimum.

The relaxation algorithm should now be able to determine the minimum energy configuration.

The unit cell lengths are simply scaled down for higher pressures investigated. In this way, the molecules have less space and more forces acting between them. This behavior simulates a material under pressure. The grid parameter values have been taken from Ref. [24].

A pressure of 4 GPa has been selected randomly. For this pressure a convergence test has been made for the unit cell. The computation result indicates that the discretization of rgkmax = ”2.5” and ngridk = ”2 3 2” is sufficient. For these values, a ground state calculation takes about 15 minutes. Doubling the above parameters results in about the same range of energy. But the computational effort increases to a whole day for a single ground state calculation.

For every function call of the BFGS method, a ground state must be calculated. So too large values of the discretization results in huge computation times. Assuming 30 calculation steps, the computational time value can vary between eight hours and one month according to the described discretization parameters.

After the calculation, the optimized atomic position are stored. For post processing, a MATLAB program has been created to compute the molecular angles based on the output file. It replicates them according to the unit cell parameters in each direction. Afterwards, the cluster algorithm as described in section 2.2 is used to condense the cloud of points into clusters of molecules.

For θ a planar plane through the atoms of the two molecules in a unit cell is fitted. Now the intersection angle between these planes is derived. For δ a straight line through each molecule is fitted to calculate the intersection angle of these lines. For χ the lines from the δ-determination can be used to determine the intersection angle with the grid vector plane $\mathbf{c}^*$.

At high pressures, the angles computed are more accurately compared to the angles computed at lower pressures. This behavior can be explained by the fact that at higher pressure, there is less space in the unit cell. This results in an other shape of the hyper surface of the energy. The gradients become higher and the minima become sharper. So the desired minimum can be reached more precisely. There is more space between the molecules in the unit cell at low pressure. The potential reacts less strongly to atomic position change. With a given accuracy there is now a larger area in which the minimum can be found.

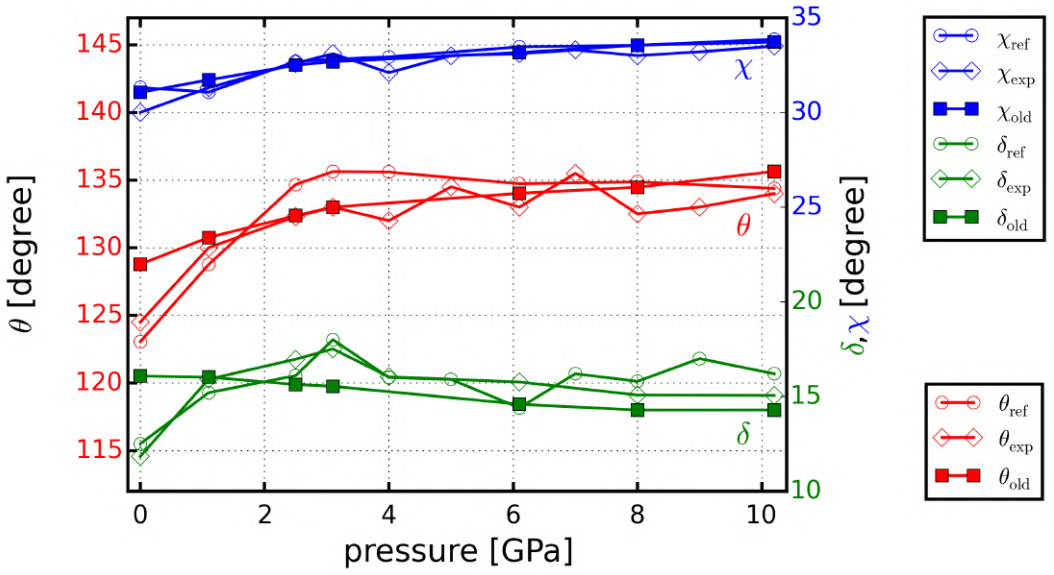

Figure 2.5.1: Comparison of the experimentally determined angles (diamond) and the calculations (circle) from Ref. [24], and the angles (square) calculated from the BFGS method.

2.6 Simplification to Rigid Molecules

While relaxing molecular crystals, two components of forces are important. First, the binding force that holds the individual atoms in the molecule and second the forces between the individual molecules, generally called van-der-Waals forces. The atomic binding force is relatively strong but localized while the van-der-Waals force has a higher interaction range. The mathematics of the BFGS algorithm, in the first iterations only considered the much bigger forces of molecule bindings. Only in a very late state of relaxation, when the force magnitude comes down to the order of the van-der-Waals forces, they also influence the computation process. So first the molecule structure is optimized and later the alignment due to other molecules is done by these much weaker forces. Sometimes, if the van-der-Waals force is weak compared to the breaking condition of the algorithm, the alignment of the molecules stays inaccurate.

For a faster relaxation, it is suggested in this work not to regard the individual atoms but only the molecules as rigid rotators. In this way, the number of degrees of freedom for a molecule is reduced to 6. In addition the internal molecular forces are ignored and only the forces between the molecules become involved in the optimization process.

The already implemented BFGS method in chapter 2.1 has now been adapted and expanded for molecules while each molecule used in the input is considered as a rigid rotator. It thus has 6 degrees of freedom, 3 translational degrees and 3 rotational degrees. If a molecule consists of only one atom due to unfavorable inputs,

only the translation coordinates are optimized, since a single atom does not need to be rotated. In this case there is mathematically no change in the description of the degrees of freedom. Nevertheless, in this work it is still considered as molecule with a single atom. In addition, other single atoms without a molecular membership are allowed. Atoms which belong to no molecule are considered as independent atoms. The calculation of independent atoms will be perform like in chapter 2.1 described.

For energy optimization using the BFGS algorithm, this means that now the optimization vector $\vec{X}$ receives another three or six entries per molecule while all degrees of freedom of the atoms that belongs to a molecule will be removed. The new added degrees of freedom are now no longer atom-bound, but molecule-wide. The N_{Atom} atoms in the lattice cell are distributed in N_{Mol} molecules. So a description of the molecule has to be defined.

A molecule is a cluster of atoms. So it is defined by the positions of its containing atoms. Since the molecule is to be regarded as a rigid rotator, its movement can be defined by the movement of a fixed molecule point and the rotation around this point. This fixed molecule point could be every arbitrary point, even random points far away from the molecule.

In particular, the rotations appear more physical when rotated around the geometric center $\mathbf{M}_m^{\mathrm{Geo}}$ or the center of mass $\mathbf{M}_m^{\mathrm{Com}}$. In this work, there is no reason to neglect the mass terms. So always the center of mass is used here and this point is generally called $\mathbf{M}_m$. However, for symmetric molecules, *e.g.*, the ancenes, the geometric center $\mathbf{M}_m^{\mathrm{Geo}}$ and the center of mass $\mathbf{M}_m^{\mathrm{Com}}$ will be the same.

$$\mathbf{M}_m^{\mathrm{Geo}} = \frac{1}{N_m} \sum_{i \in m} \mathbf{r}_i^{k=0} \text{ , or} \tag{2.6.1}$$

$$\mathbf{M}_m^{\mathrm{Com}} = \frac{\sum_{i \in m} m_i \mathbf{r}_i^{k=0}}{\sum_{i \in m} m_i} \text{ ,} \tag{2.6.2}$$

where $N_m = \sum_{i \in m} 1$ stands for the number of atoms in the molecule m. In this work, indices i always refer to atoms and indices m always refer to molecules. Here, the fixed start positions $\mathbf{r}_i^{k=0}$ of the atoms are used as specified in the input file. This starting center $\mathbf{M}_m$ becomes the initial point of the translation of the whole molecule m and it stays constant for the whole relaxation.

The translation is simply the displacement of the molecule based on the starting center of the molecule $\mathbf{M}_m$ along the respective axes by the degrees of freedom $\mathbf{T}_m = (T_{x_m}, T_{y_m}, T_{z_m})^T$. Since the starting center of the molecule is known, $T_{x_m}^{k=0} = T_{y_m}^{k=0} = T_{z_m}^{k=0} = 0$ is defined as starting value of the displacement. Thus the exact

atomic translational position of an atom i belonging to molecule m is

$$\mathbf{r}_{i\in m}^{trans}(\mathbf{T}_m) = \mathbf{r}_i^{k=0} + \mathbf{T}_m \ . \tag{2.6.3}$$

The rotation is slightly more complicated. The easiest way to perform rotations is around the origin of a coordinate system. Therefore, a linear coordinate transformation into the molecule coordinate system where the midpoint $\mathbf{M}_m$ is at the origin will be performed. Now, the rotations can be executed around the coordinate axes x, y and z and finally, the result must be transformed back to the starting coordinate system. The starting value of the degrees of freedom is thus $\alpha_m^{k=0} = \beta_m^{k=0} = \gamma_m^{k=0} = 0$ and these degrees of freedom are measured as the angles between the starting configuration and any step positions.

This can be described with the following position formula of an atom i at the position $\mathbf{r}_i$ which can be easily extended by the translation description 2.6.3.

$$\mathbf{r}_{i\in m} = \hat{R}_z(\gamma_m)\hat{R}_y(\beta_m)\hat{R}_x(\alpha_m)\left(\mathbf{r}_i^{k=0} - \mathbf{M}_m\right) + \mathbf{M}_m + \mathbf{T}_m(T_x, T_y, T_z) \tag{2.6.4}$$

$$\mathbf{r}_{i\in m} = \mathbf{r}_i(T_{x_m}, T_{y_m}, T_{z_m}, \alpha_m, \beta_m, \gamma_m) \ , \tag{2.6.5}$$

where $\hat{R}_{\{xyz\}}(\omega)$ are the rotation matrices for rotation by the angle ω along the corresponding three coordinate axis:

$$\begin{aligned}
\hat{R}_x(\alpha) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad \hat{R}_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \\
\hat{R}_z(\gamma) &= \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \ .
\end{aligned} \tag{2.6.6}$$

For atoms i belonging to molecule m n exact position formula is set up now. This position formula contains only the degrees of freedom of the molecule and some predefined constants that vary from molecule to molecule but are constant over the relaxation. The constants describe the starting position of the atoms $\mathbf{r}_{i\in m}^{k=0}$ and hence the resulting center position $\mathbf{M}_m$. For each calculation step when a new position trial is calculated, the same starting conditions are applied to the atoms and then the degrees of freedom suggested by the BFGS routine can be applied to obtain the new position of the atoms.

Analogous to the case without molecules, the energy and its derivatives are used for the optimization according to the degrees of freedom. Formally, the total energy

E is now determined as follows:

$$E(\vec{X}) = E(\{x_i\},\{y_i\},\{z_i\},\{T_{xm}\},\{T_{y_m}\},\{T_{zm}\},\{\alpha_m\},\{\beta_m\},\{\gamma_m\}) \\ i = \{1,2,...,N_{\text{independent}}\} \\ m = \{1,2,...,N_{\text{Mol}}\}\,. \tag{2.6.7}$$

Here, x_i, y_i, z_i describe the degrees of freedom of the independent atom i. These are the atoms which do not belong to a molecule. $T_{x_m}, T_{y_m}, T_{z_m}$ describes the translations and $\alpha_m, \beta_m, \gamma_m$ the rotation of the molecule m. In total, $3N_{\text{independent}}+6N_{\text{Mol}}$ degrees of freedom were used in the optimization when the full problem is assumed. The ordering of the coordinates in the vector $\vec{X}$ does not matter but has to be fixed for a complete BFGS calculation.

For the new degrees of freedom, formally the gradient g is:

$$\vec{g} = \vec{\partial}E = \\ \left(\left\{\frac{\partial E}{\partial x_i}\right\},\left\{\frac{\partial E}{\partial y_i}\right\},\left\{\frac{\partial E}{\partial z_i}\right\},\left\{\frac{\partial E}{\partial T_{x\,m}}\right\},\left\{\frac{\partial E}{\partial T_{y_m}}\right\},\left\{\frac{\partial E}{\partial T_{z\,m}}\right\},\left\{\frac{\partial E}{\partial \alpha_m}\right\},\left\{\frac{\partial E}{\partial \beta_m}\right\},\left\{\frac{\partial E}{\partial \gamma_m}\right\}\right)^T . \tag{2.6.8}$$

The derivatives for the single atom coordinates x_i,y_i and z_i remain unchanged. It remains the negative force acting on this single atom i as shown in formula 2.1.5. These coordinates and gradients are assigned to all the atoms that do not belong to a molecule.

The determination of the molecular gradients is done by using the exact atom positions of the atoms belonging to a molecule in each calculation step. In this way, the formula 2.6.4 can be used to calculate the gradient of the total energy E. For any molecular-related degrees of freedom $f_m \in [T_{x_m}, T_{y_m}, T_{z_m}, \alpha_m, \beta_m, \gamma_m]$, the following formula applies:

$$\frac{\partial E}{\partial f_m} = \sum_{i\in m} \frac{\partial E}{\partial \mathbf{r}_i}\frac{\partial \mathbf{r}_i}{\partial f_m} = -\sum_{i\in m} \mathbf{F}_i \frac{\partial \mathbf{r}_i}{\partial f_m}\,. \tag{2.6.9}$$

For the energy gradients only the partial derivatives of the positions according to the respective degrees of freedom are needed. For translation, the derivative of the position according to the translational degrees of freedom is given exactly as one:

$$\frac{\partial E}{\partial \mathbf{T}_m} = \sum_{i\in m} \frac{\partial E}{\partial \mathbf{r}_i}\frac{\partial \mathbf{r}_i}{\partial \mathbf{T}_m} = -\sum_{i\in m} \mathbf{F}_i \frac{\partial \mathbf{r}_i}{\partial \mathbf{T}_m} = -\sum_{i\in m} \mathbf{F}_i\,. \tag{2.6.10}$$

Thus, the molecular translation, analogous to individual atoms, is determined from the sum of the forces acting on the atoms belonging to the molecule m along the respective axes.

For the rotation, the derivative of the standard rotation matrices according to

the angles α_m, β_m and γ_m are derived. To make the following formulas smaller, s and c will be shortcuts for the sine and cosine terms, followed by the angles as argument.

$$\begin{aligned}
\frac{\partial E}{\partial \alpha_m} &= -\sum_{i \in m} \mathbf{F}_i \cdot \mathbf{A}_i(\alpha_m, \beta_m, \gamma_m) \\
\frac{\partial E}{\partial \beta_m} &= -\sum_{i \in m} \mathbf{F}_i \cdot \mathbf{B}_i(\alpha_m, \beta_m, \gamma_m) \\
\frac{\partial E}{\partial \gamma_m} &= -\sum_{i \in m} \mathbf{F}_i \cdot \mathbf{C}_i(\alpha_m, \beta_m, \gamma_m) \text{ , with}
\end{aligned} \tag{2.6.11}$$

$$\mathbf{A}_i(\alpha,\beta,\gamma) = \begin{pmatrix} s(\alpha)\left[-r_{iz}c(\gamma)s(\beta) + r_{iy}s(\gamma)\right] + c(\alpha)\left[r_{iy}c(\gamma)s(\beta) + r_{iz}s(\gamma)\right] \\ c(\alpha)\left[-r_{iz}c(\gamma) + r_{iy}s(\beta)s(\gamma)\right] - s(\alpha)\left[r_{iy}c(\gamma) + r_{iz}s(\beta)s(\gamma)\right] \\ c(\beta)\left[r_{iy}c(\alpha) - r_{iz}s(\alpha)\right] \end{pmatrix}$$

$$\mathbf{B}_i(\alpha,\beta,\gamma) = \begin{pmatrix} c(\gamma)\left[r_{iz}c(\alpha)c(\beta) + r_{iy}c(\beta)s(\alpha) - r_{ix}s(\beta)\right] \\ s(\gamma)\left[r_{iz}c(\alpha)c(\beta) + r_{iy}c(\beta)s(\alpha) - r_{ix}s(\gamma)\right] \\ -r_{ix}c(\beta) - s(\beta)\left[r_{iz}c(\alpha) + r_{iy}s\alpha\right] \end{pmatrix}$$

$$\mathbf{C}_i(\alpha,\beta,\gamma) = \begin{pmatrix} r_{iz}c(\gamma)s(\alpha) - s(\gamma)\left[r_{ix}c(\beta) + r_{iy}s(\alpha)s(\beta)\right] - c(\alpha)\left[r_{iy}c(\gamma) + r_{iz}s(\beta)s(\gamma)\right] \\ r_{ix}c(\beta)c(\gamma) + c(\alpha)\left[r_{iz}c(\gamma)s(\beta) - r_{iy}s(\gamma)\right] + s(\alpha)\left[r_{iy}c(\gamma)s(\beta) + r_{iz}s(\gamma)\right] \\ 0 \end{pmatrix}, \tag{2.6.12}$$

where $\tilde{\mathbf{r}}_i = \mathbf{r}_i^{k=0} - \mathbf{M}_m = (r_{ix}, r_{iy}, r_{iz})^{\mathrm{T}}$ are the molecular coordinates of the atom $i \in m$.

Now, for each molecule, the gradients according to the degrees of freedom can be calculated. So all required inputs for the energy minimum search using the BFGS algorithm have been set up now.

The so modified gradients are no longer the exact representation of the forces acting on the atoms. And hence the termination condition used before becomes invalid. But the complete gradient should always converge to zero. A possible termination condition is now, that the sum of the gradients divided by the number of gradients falls below a certain value.

Since the molecule in this method is a rigid rotator, however, it must be sure that the input structure of a molecule is already a relaxed structure. If this condition is not given, then the structure relaxation could not be reliable. So the creation of properly relaxed molecules before start of the calculation in the crystal is essential. With **`exciting`** calculation of individual molecules is possible. To do this, a super cell with the desired molecule is creates and filled with sufficient vacuum. In this way, no interaction between the replicated crystal images occurs. Even if a crystal

is calculated, the individual lattice points do not interact with each other and the energy depends on the desired molecule only.

This calculation works with uncharged molecules only. The Coulomb range of charged particles is simply to large and the computational time goes rapidly up with the increased vacuum. In the future, this could be prevented by the Coulomb cutoff interaction.

To simplify the relaxation of already given crystals, a Python script has been written. This script can read an input file and receive a clustering parameter from the user as described in chapter 2.4. Thus, the atoms from the unit cell are combined into molecules. Found molecules are rewritten into separate input files and provided with a parameter for vacuum. These input files can be used within an **exciting** calculation which optimizes the molecule to a complete relaxed structure.

The results of the **exciting** calculation can be returned to the Phyton script again. Then, the script collects all optimized structure data from the previous calculations and inserts them back into a new input file containing the originally structure with exchanged molecules.

The molecular membership of the atoms is not calculated by **exciting** and must be given by a input file. The molecular membership could be set manually or by the Phyton script.

The recognition of identical molecules is not implemented now in the Phyton script. In the case of anthracene, two identical molecules are present in a unit cell. So a single calculation in vacuum is sufficient to relax the molecule and use this information twice.

When writing the new input file, the atomic position of a molecule has been changed by an **exciting** calculation. Both, the translational and rotational infora-tion of the molecules, are lost. This information must be restored as good as possible to prevent muffin-tin overlaps. An exact mapping of the old molecule and the relaxed molecule does not exist because the relaxation will shift the atomic positions. The center of each molecule can be calculted from the original input file as well as from the **exciting** output data. So the translational information can be restored by simply superposing the corresponding molecular centers. For the rotation the angular deviations between each old and newly relaxed atom are calculated. The molecule is then rotated to a position with a minimal mean angle deviation. The rotation need not been as exact as possible, since the following relaxation process will correct any rotation deviation left. In this way, a new molecule structure is found which is as closed as possible to the original input structure but with completely relaxed molecules. But it is recommended to check the machine generated

input files to avoid muffin-tin overlaps on start manually.

The method of the rigid rotator has now been tested on the molecular crystal anthracene using the above preparations.

For this purpose, the molecules had to be identified by a clustering algorithm or by hand. Each molecule should be assigned to a molecular membership mol = {1,2} in the input file.

To calculate the center of the molecule correctly, the atoms of the molecule must be set to form the fully visible molecule even if they are no longer within the unit cell. As explained in the section 2.2 otherwise a wrong molecular center will be applied.

In addition, **exciting** has a function to automatically translate all atoms located outside back into the unit cell. However, this is done before the center of mass of the molecule $\vec{M}_m$ can be determined. In this way, despite correct positioning of the molecular atoms, it results in an incorrect center of the molecule. To calculate a correct molecular center, it is important here to turn off this behavior by setting the parameter "`tshift`" to "`false`". However, the parameter "`tshift`" is only apply with Cartesian coordinates, so nevertheless no internal coordinates can be used here. Theoretically the algorithm can handle internal coordinates but by using atom positions outside the lattice cell, a false center of mass will be calculated.

To relax all the existing molecules as rigid rotators using the BFGS method the input parameter "`rigid`" must be set to "`true`".

To minimize the energy of an anthracene crystal which has two molecules in the unit cell, there are 12 degrees of freedom present. Each of these degrees of freedom gets its own parameterization (coordinates). Also every coordinate gets a gradient value for each calculation step assigned. This assignment is controlled internally in the program and may vary for each input. The results of a test calculation are shown in figure 2.6.1. In this case, the algorithm reserves 6 degrees of freedom per molecule, where the first 3 are the translations in x, y and z direction and the last three are the rotations around the 3 coordinate axes. In the picture 2.6.1, *e.g.*, g7 describes the translation coordinate of the second molecule along the y axis. Depending on whether there are still individual atoms without molecular affiliation present, the list of degrees of freedom can also be reordered.

The Figure 2.6.1 shows, that the method has found a minimum. It is visible that the energy decreases continuously every step of the calculation and the gradients of each step of the calculation went to zero.

The shift of the parameterized coordinates from the starting position describes

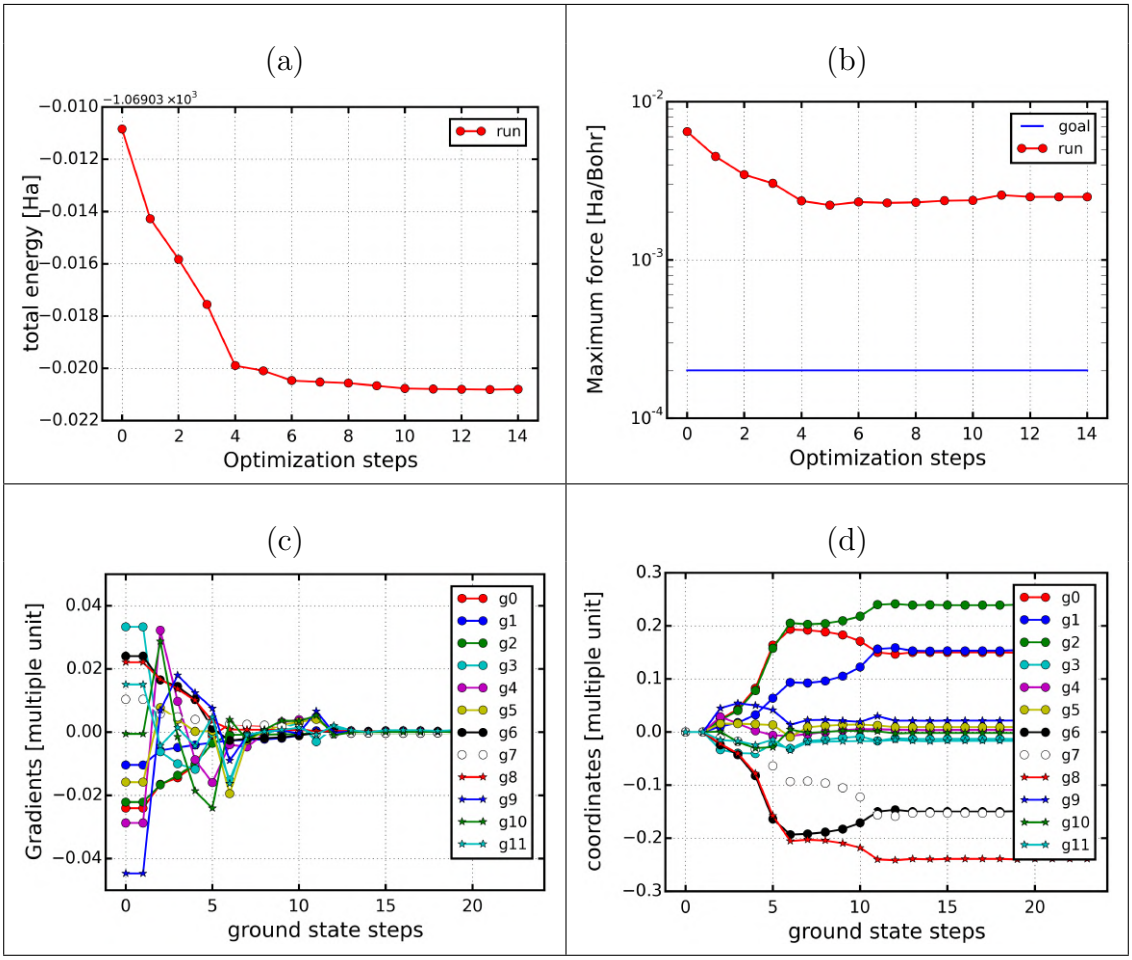

Figure 2.6.1: Relaxation of anthracene crystal where the molecules are considered to be rigid rotators. Therefore only 6 degrees of freedom per molecule are used. The total energy decreases continuously with each calculation step (a) and the gradients went to zero (c). So a local minimum in the considered energy space was found. But the gradients do not correspond to the force between the atoms anymore. And since not all degrees of freedom of the system have been considered, there remain force components between the atoms (b) which do not go to zero. Using this method the maximum force reduction is limited and can not fall under a certain limit. The energy and the maximum force (a,b) were shown in BFGS computational steps while the gradients (c) and coordinates as parameterized degrees of freedom (d) are shown in ground state computation steps. These steps are different in the line search.

the movement of the molecules and their rotation during the optimization process.

However, even though the maximum force component between the atoms is reduces in the beginning, it approaches a constant level which is higher than the goal. Thus, with this calculation, the desired force relaxation is not achieved. It is visible in Figure 2.6.1 that not all forces between the atoms will disappear. This fact can be dedicated to the reduction of the degrees of freedom. Within the rigid rotator method, the molecular positions can be determined quite accurately against each other. But since each molecule is regarded as rigid, the individual force components within a molecule can not be balanced. The distances of the atoms within the molecule must always remain the same. Although the molecules should have no internal forces at start, the presence of other atoms or molecules induced additional internal forces that can not be relaxed by the algorithm.

In the figure 2.6.1, the energy and maximum force were plotted only every BFGS calculation step, while the gradients and coordinates are also shown at each line search step within the BFGS steps. Thus, the gradients show the exact number of function calls, here the ground state calculations, while the energy shows the number of BFGS operations.

2.7 All molecular Degrees of Freedom

The handling of the molecules as rigid rotators has the disadvantage that the molecules are not relaxed. A molecule, which is individually relaxed in vacuum, receives additionally forces due to the presence of other atoms or molecules in the unit cell. These new forces causes a deformation of the molecules. Even if this effect is only slightly, a different molecule structure occurs.

Now forces between the atoms in the molecule appear, which of course can not be changed by displacement or rotation of the whole molecule. Using the concept of rigid rotators, the minimum of the energy can not be found precisely because it does not allow the atoms in the molecule to move against each other. Only the alignment of the molecules with each other can be optimized.

One possible approach would be to run the calculation in a sequence. First the molecular orientation is determined with the rigid rotator approach and then the atoms in the molecules are optimized using a normal relaxation calculation. However, these steps should be used in an iterative algorithm in alternate mode. Molecular optimization can drastically change the molecular orientations against each other and this orientation can influence the atomic positions in the molecule structure.

To find the minimum of the energy more degrees of freedom for the molecules must be considered. A naive approach would be to include all atomic positions as degrees of freedom in the standard BFGS algorithm and in addition the 6 molecular degrees of freedom per molecule. In this way, a system with N_{Mol} molecules is considered consisting of a total of N_{Atom} atoms. In total, there are $3N_{\mathrm{Atom}}$ degrees of freedom pressent. However, the naive approach will consider exactly $3N_{\mathrm{Atom}}+6N_{\mathrm{Mol}}$ degrees of freedom, resulting in an over determined system. *E.g.*, the translation in the x direction of an atom would be determined by both T_x^m and x_i. Tests with this configuration have shown that the convergence of the algorithm decreases significantly as the individual degrees of freedom starts to oscillate. This is an undesired behavior, as it drives the number of ground state calculations extremely high and thus means very long computation times. So it is reasonable to use exactly $3N_{\mathrm{Atom}}$ degrees of freedom for relaxation, no more and no less.

So a different approach for the degrees of freedom of a molecule m, which consists of N_m atoms is required. Any molecule m has exactly $3N_m$ degrees of freedom which must be described first. In section 2.6 already 3 translational degrees of freedom and 3 rotational degrees of freedom have been considered. There are still $(3N_m-6)$ degrees of freedom left called vibration in general. Molecules that are built up in an exact linear manner, *e.g.*, CO_2 have one rotational degree of freedom less because the rotation around the molecular axis is invariant here. Nevertheless, the molecule generally has $3N_m$ degrees of freedom, so for such a molecule, one more vibrational degree of freedom exist $(3N_m-5)$. This is not important for rigid molecules, as there is always a degree of freedom reduction if there are more than two atoms in the molecule. So a description of the vibrational states of the molecules is required.

Performing a complete vibration analysis for each individual molecule, however, becomes expensive. It would be necessary to determine the spring constants of the atoms relative to each other in order to be able to build up the dynamic matrix of the molecule [31]. In this way, the eigenvectors are obtained as a vibration basis. To achieve this, each atom would have to be individually deflected by a distance Δ in each spatial direction in order to determine the force differences to the deflection position. This requires $3N_m$ ground state calculations per molecule m without actually getting started on structure optimization. If the unit cell consists of molecules only, then a maximum of $3N$ calculations have to be made. Commonly, these are more ground state calculations than desired in comparison to the number used for structure optimization. So this is not the best way to proceed.

That is why this work proposes a different way. In order to get a description of the vibrational states, a basis transformation of the degrees of freedom is set up. At start, the standard basis of the molecular degrees of freedom is used where each

atom can be individually deflected in x, y and z direction. The position information of the atoms in the molecule m can be described with a position vector $\vec{x}_m^{\text{total}}$:

$$\vec{x}_m^{\text{total}} = (\mathbf{r}_1^{\mathrm{T}}, \mathbf{r}_2^{\mathrm{T}}, \cdots)^{\mathrm{T}} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ \vdots \end{pmatrix} . \tag{2.7.1}$$

And the degrees of freedom are the coefficients of the basis vectors $\vec{e}_a$:

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \vec{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \vec{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \vec{e}_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \cdots . \tag{2.7.2}$$

The position vector $\vec{x}_m^{\text{total}}$ has $3N_m$ entries, so the total dimension of the basis is also $3N_m$. In the unit cell, for every molecule this basis can be defined separately. To simplify the method setup, one molecule m is considered here only, but the transformation has to be done for all molecules present.

Now another basis for the molecule m is set up, called molecule basis. In this molecule basis there are 3 basis vectors for translation, 2 or 3 basis vectors for rotation, and all the remaining basis vectors for the vibrations. Both, the standard basis and the molecule basis are descriptions of the position vector $\vec{x}_m^{\text{total}}$.

The 3 translation vectors are intuitively known. To shift a whole molecule, each atom of this molecule have to be shifted in the same direction with identical distances. This can be split up in Cartesian coordinates:

$$\vec{E}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \vec{E}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} \vec{E}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ \vdots \end{pmatrix} . \tag{2.7.3}$$

So, *e.g.*, the first vector shifts all atoms of the considered molecule m along the x-direction.

The computation of the rotation vectors is not as simple as the translation vectors. The rotation of a molecule can, *e.g.*, be defined by rotary matrices. Then this description is a matrix and not a vector. In order to get a vector again, an infinitesimal rotation approach is used. These rotation vectors are derived from the eigenvectors of the inertial tensor of the molecule as in Gaussian [32] described. But since the rotation from Gaussian is used, there exist conventions. In Gaussian all coordinates are mass weighted:

$$\mathbf{q}_i = \sqrt{m_i}\mathbf{r}_i \ . \tag{2.7.4}$$

Later calculations (as shown in appendix A) indicate that the convention of Gaussian have to be canceled as soon as possible since the mass weighted coordinates have a huge influence on the BFGS-routine.

The inertial tensor $\hat{I}$ becomes:

$$\hat{I}_m = \begin{pmatrix} \sum_{i\in m} m_i(y_i^2+z_i^2) & -\sum_{i\in m} m_i x_i y_i & -\sum_{i\in m} m_i x_i z_i \\ -\sum_{i\in m} m_i y_i x_i & \sum_{i\in m} m_i(x_i^2+z_i^2) & -\sum_{i\in m} m_i y_i z_i \\ -\sum_{i\in m} m_i z_i x_i & -\sum_{i\in m} m_i z_i y_i & \sum_{i\in m} m_i(x_i^1+y_i^2) \end{pmatrix} \ . \tag{2.7.5}$$

The intrinsic values of the inertial tensor are the principal moments. But here the 3 eigenvectors are of interest. These eigenvectors are combined into the matrix $\widehat{G}_{3\times 3}$ which is required to set up the rotation basis vectors.

The rotation basis vectors are determined as follows:

$$\begin{aligned} E_{4\nu i} &= \left[(P_y)_i\widehat{G}(\nu,3) - (P_z)_i\widehat{G}(\nu,2)\right]/m_i \\ E_{5\nu i} &= \left[(P_z)_i\widehat{G}(\nu,1) - (P_x)_i\widehat{G}(\nu,3)\right]/m_i \\ E_{6\nu i} &= \left[(P_x)_i\widehat{G}(\nu,2) - (P_y)_i\widehat{G}(\nu,1)\right]/m_i \ . \end{aligned} \tag{2.7.6}$$

Where P is the scalar product of the position vector $\tilde{\mathbf{r}}_i = \mathbf{r}_i^{k=0} - \mathbf{M}_m$ in molecular coordinates of the i-th atom and the respective row of the eigenvector matrix $\widehat{G}$. The term $\nu = \{1,2,3\}$ describes the coordinates x, y and z. In this way, the three $3N_m$ dimensional rotation vectors along with the N_m atoms numbered with $i \in m$ are given.

In the source, however, the rotation vectors entries have to be divided by the square root of the atom mass, since a mass term is already known from the inertial tensor and thus its eigenvectors and to match their convention of mass weighted

coordinates. This work do not use the mass weighted coordinates. So here, it was divided by the whole mass term, unlike in the source.

E.g., the basis vector 4 is defined as follows:

$$\vec{E}_4 = \begin{pmatrix} \left[\left(\tilde{\mathbf{r}}_1 \cdot \widehat{G}(:,2)\right)\widehat{G}(1,3) - \left(\tilde{\mathbf{r}}_1 \cdot \widehat{G}(:,3)\right)\widehat{G}(1,2)\right]/m_1 \\ \left[\left(\tilde{\mathbf{r}}_1 \cdot \widehat{G}(:,2)\right)\widehat{G}(2,3) - \left(\tilde{\mathbf{r}}_1 \cdot \widehat{G}(:,3)\right)\widehat{G}(2,2)\right]/m_1 \\ \left[\left(\tilde{\mathbf{r}}_1 \cdot \widehat{G}(:,2)\right)\widehat{G}(3,3) - \left(\tilde{\mathbf{r}}_1 \cdot \widehat{G}(:,3)\right)\widehat{G}(3,2)\right]/m_1 \\ \left[\left(\tilde{\mathbf{r}}_2 \cdot \widehat{G}(:,2)\right)\widehat{G}(1,3) - \left(\tilde{\mathbf{r}}_2 \cdot \widehat{G}(:,3)\right)\widehat{G}(1,2)\right]/m_2 \\ \left[\left(\tilde{\mathbf{r}}_2 \cdot \widehat{G}(:,2)\right)\widehat{G}(2,3) - \left(\tilde{\mathbf{r}}_2 \cdot \widehat{G}(:,3)\right)\widehat{G}(2,2)\right]/m_2 \\ \left[\left(\tilde{\mathbf{r}}_2 \cdot \widehat{G}(:,2)\right)\widehat{G}(3,3) - \left(\tilde{\mathbf{r}}_2 \cdot \widehat{G}(:,3)\right)\widehat{G}(3,2)\right]/m_2 \\ \vdots \end{pmatrix}. \tag{2.7.7}$$

The double points mean that the entire line of the matrix is used as a vector.

At this point, linear molecules can be easily identified here. If a molecule is linear, then the scalar product of the rotation basis vectors with itself becomes nearly zero. Without rounding error, the result of a linear molecule would become exactly zero. If it is not linear, three usable rotation basis vectors can be determined. So, comparing the rotation vectors norms to zero, a decision is possible, whether more or less vibration basis vectors are required.

The molecular basis so far, consists of the first 5 or 6 basis vectors and must be filled up to the required dimension $3N_m$. Since no exact vibration analysis is desired here it is not necessarily to compute the exact vibration modes. It is enough if the molecular basis spans the entire $3N_m$ dimensional space and fits the two following conditions. First, the vibration basis vectors must not move the molecule in space. This means that the molecular center of mass $\mathbf{M}_m$ does not change. This condition is contained in the stricter condition that the basis vectors must be orthogonal to each another, since the translation vectors already cover the complete translational subspace. Second, at least one vibration basis vector must have one non-zero entry for each directional component of an atom. This is important to make the complete $3N_m$ dimensional space accessible to the BFGS method.

Here is a example for existence of the second condition: A molecule with the 3 translation basis vectors and 3 rotation basis vectors is given. All other basis vectors of the vibration have zero entries in the first component which describing the x position of the first atom. Theoretically, this could describe the complete $3N_m$ dimensional space. But the BFGS method can now move the first atom in the

x-direction only through the 3 translation basis vectors and 3 rotation basis vectors. Any movement of this atom in the x direction by vibrations is prevented by the zero entries of the vibration basis vectors. The correct positioning of the atom is only possible by movement of all atoms which limits the scope of action for the BFGS method and which can even be blocked by muffin-tin radii.

Since no other conditions on the vibration basis are required these basis vectors can be filled by random numbers. Ideally, there is a very simple algorithm that creates more orthogonal vectors for given orthogonal basis vectors, the Gram-Schmidt algorithm [33]. Since the first orthogonal basis vectors are already known, random vectors can be orthogonalized until the molecular basis is complete. Other algorithms for orthogonalization such as the Givens rotation[34] require an already prepared basis set. Then, this basis set is orthogonalized. However, the already found translation basis vectors and rotation basis vectors will be destroyed in this way. Hence the usefulness of the complete basis will be complete destroyed. Gram-Schmidt leaves already found orthogonal vectors and allows the addition of further orthogonal vectors. However, it is important to keep in mind that the original algorithm is numerically extremely unstable, since the rounding errors due to the floating point representation in the computer sum up with each new basis vector. For larger basis, the basis will be no longer orthogonal anymore using this algorithm. However, the modified Gram-Schmidt algorithm [35] exists, which is not afflicted with this problem and is therefore implemented here.

Random numbers are used as vibration modes and therefore they have to be called pseudo-vibration modes. For simple molecules like Cl_2, the vibration modes thus found match the real vibration modes found in an analytic way. For more complex molecules, however, the computed vibration modes can also be any linear combination of the analytic vibration modes.

For each molecule in the unit cell, a pseudo-vibration basis can be created. This basis spans the entire $3N_m$ dimensional space for the configuration $\vec{x}_m^{\text{total}}$ of the molecule. However, for the evolution of the system in the form of BFGS steps, the infinitesimal rotation basis vectors may change. So the created vibration basis is correct only for the start configuration. A new suitable vibration basis has to be created for each further BFGS step. The computation time is not expensive but for the BFGS method this way is problematic.

The BFGS method has a memory of previous calculated steps. The information about the last positions and gradients are stored in the approximate Hessian matrix and is included in the prediction of the minimum for each new calculation. If, however, the vibration basis vector changes in every step of the calculation, then this storage of the last but no longer up-to-date information of the vibration basis

vector will flow into the new prediction. This will lead to false results. For each calculation the setup of new degrees of freedom is used but the old ones still influence the computation. The vibration basis must therefore not be changed from step to step in order to not change the definition of the used degrees of freedoms.

To avoid a permanent recalculation of the vibration basis, the starting configuration is used for the basis transformation only and will be rotated according to the rotation of the molecule.

The position formula of the atoms of a molecule can therefore be easily extended:

$$\begin{aligned} \tilde{\mathbf{r}}_i &= \mathbf{r}_i^{k=0} - \mathbf{M}_m + v_1\mathbf{V}_1 + v_2\mathbf{V}_2 + \cdots + v_{3N_m-6}\mathbf{V}_{3N_m-6} \\ \mathbf{r}_{i\in m} &= \hat{R}_z(\gamma)\hat{R}_y(\beta)\hat{R}_x(\alpha)\left[\tilde{\mathbf{r}}_i\right] + \mathbf{M}_m + \mathbf{T}_m(T_x, T_y, T_z) \end{aligned} \tag{2.7.8}$$

$$\mathbf{r}_{i\in m} = \mathbf{r}_{i\in m}(T_x, T_y, T_z, \alpha, \beta, \gamma, v_1, v_2, \cdots, v_{3N_m-6})\ . \tag{2.7.9}$$

The position of an atom i from the molecule m thus depends on all $3N_m$ degrees of freedom.

The center of the molecule is set to the coordinate origin due to a coordinate transformation to the molecular coordinate system. Then the atoms get deflected from their starting positions according to the vibration degrees of freedom. Next the whole molecule is rotated around all 3 space coordinate axes. Then the back transformation is performed to the general coordinate system specified by the unit cell. Now, the entire molecule is translated according to the translational degrees of freedom along the three spatial directions. The result of this procedure describes the positions $\mathbf{r}_{i\in m}$ of each atom i from the molecule m. This is true in any given calculation step providing the degrees of freedom $(T_x, T_y, T_z, \alpha, \beta, \gamma, v_1, v_2, \cdots, v_{3N_m-6})$ from the BFGS algorithm.

When the vibration basis is set up, the translation vectors and rotation vectors can be omitted. The rotation is further described by the rotation matrices, which is much more accurate for larger rotations than the infinitesimal rotation vectors. The translation is indirectly defined again by the shift of the molecular center of mass. So the vibration basis vectors for the atom positions must be stored only.

The vibration vectors $\vec{V}_1, \vec{V}_2, \cdots$ are actually $3N_m$ dimensional. For an atom, however, only the respective x_i, y_i and z_i components of the corresponding atom i out of the considered vibration vector $\vec{V}_\mu$ are important. This reduces the representation of the vibration basis vectors to the corresponding three-vector $\mathbf{V}_{\mu,i}$ for the respective atom i.

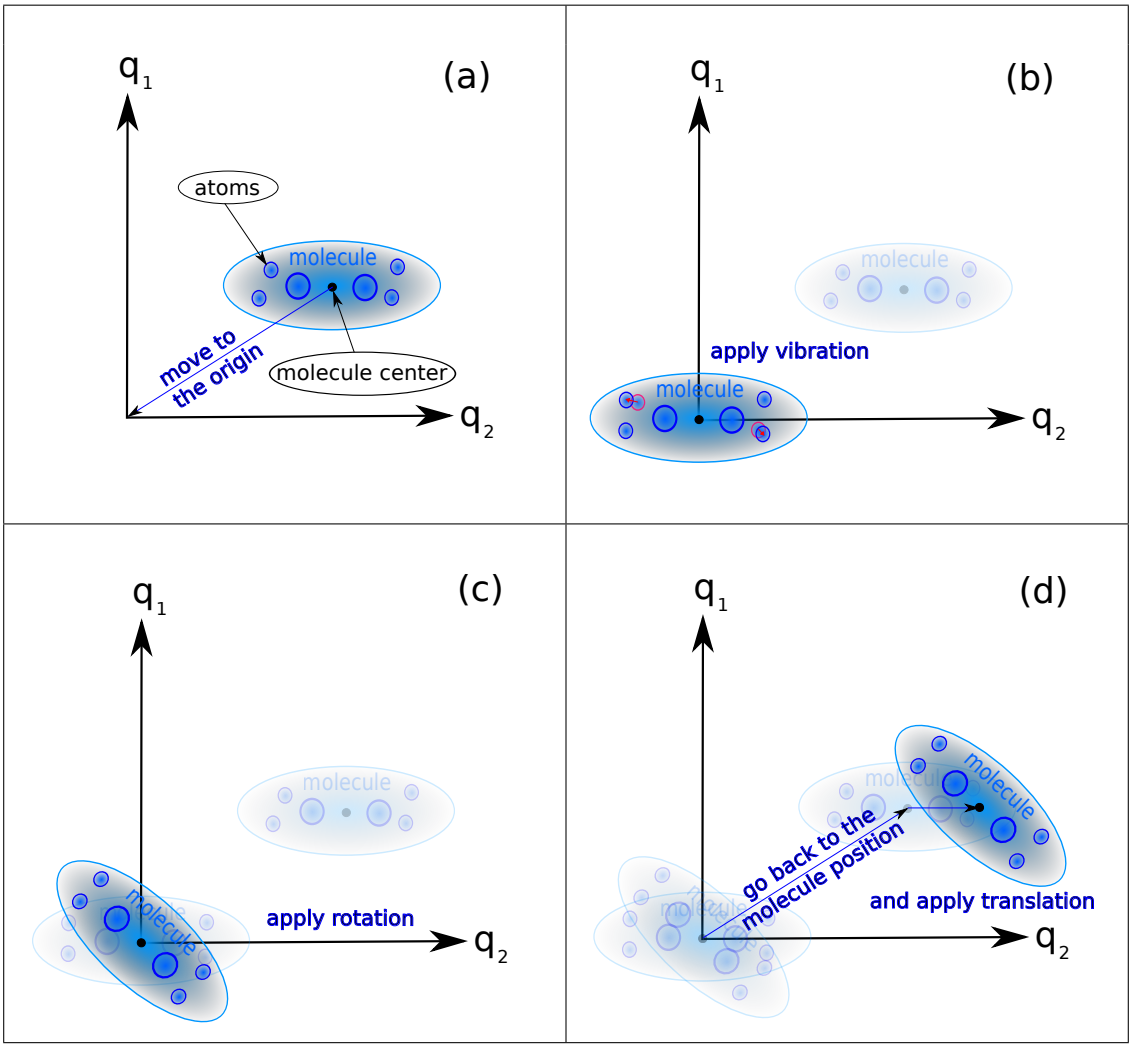

Figure 2.7.1: Visualizing of the position formula 2.7.8. First, the initial values of the atoms are loaded (a). Thereafter, they are shifted to the coordinate origin corresponding to the molecule center point (a). There, first the vibration (b) and then the rotation (c) is applied. Then the atoms are set back to the original position of the molecule (d) and then the translation is applied (d). This process depends only on the starting conditions of the molecule and the degrees of freedom of translation, rotation and vibration of this one molecule.

The calculation of the gradients is now based again on the formula 2.6.9. This also allows the calculation of the gradients of the new vibration degrees of freedom because they are molecular variables and the position formula 2.7.8 depends on this variables.

The translation gradient remains unchanged by this method. In deriving the position formula according to the translational degrees of freedom, the complete rotation term is omitted because it does not depend on translational degrees of freedom. The rotational gradient is changed minimal only. The molecule coordinate $\tilde{\mathbf{r}}_i$ (according Eq. 2.7.8) now also contains the complete vibration terms. However, this does not have any relevance to the rotation derivation, since $\tilde{\mathbf{r}}_i$ does not depend on rotational degrees of freedom. The rotation gradient thus remains formally the same with a modified $\tilde{\mathbf{r}}_i$ term.

The derivation of the position due to a vibration degree of freedom v_ν is:

$$\frac{\partial E}{\partial v_\nu} = \sum_{i \in m} \frac{\partial E}{\partial \mathbf{r}_i} \frac{\partial \mathbf{r}_i}{\partial v_\nu} = -\sum_{i \in m} \mathbf{F}_i \cdot \left[\hat{R}_z(\gamma) \hat{R}_y(\beta) \hat{R}_x(\alpha) \mathbf{V}_\nu \right], \tag{2.7.10}$$

and thus the scalar product of the force and the vibration basis three-vector which is co-rotated with the molecule m for each atom i from the considered molecule.

To implement this algorithm one issue with the number of degrees of freedom in linear molecules is left. Theoretically, for linear molecules, there are $3N_M+1$ degrees of freedom now. The 3 degrees of freedom for the translation, the 2 rotational degrees of freedom, the $3M_n-5$ vibrational degrees of freedom, and the last rotational degree of freedom, which was not subtract.

In theory, this does not pose a problem because a rotation subspace can be defined in the space of the dimension $3N_m$, which itself contains 3 degrees of freedom. This subspace is only 2 dimensional in the case of a linear molecule, but can be described with 3 variables, which then depend on each other. However, this dependence is limited to the rotational subspace and there is no correlation with translation and vibration if the basis is chosen to be sufficiently orthogonal. Correlations are therefore in the rotation subspace only. But since the 3 rotations in the real space are orthogonal, the expectation is that the mapping from the 2D-dimensional rotation subspace to the 3 dimensional space is one-to-one and therefore no further correlations occur and the rotation does not start to oscillate. Eventually a rotational degree of freedom will always be 0.

In the practical implementation, however, it was unfortunately that the exact number of degrees of freedom where not known until after initialization, because the size of some variables such as the location vector must be known for the program before they can be filled. Therefore, a different rotation subspace for linear molecules

was implemented. A subspace for rotation which has only two degrees of freedom. In this way, the molecule description is always done with exactly $3N_m$ guaranteed degrees of freedom.

For this it makes sense to use the polar representation, which works on its own with two angles ϕ and θ. However, since these are curvilinear coordinates, one can not simply apply $\phi + \Delta\phi$ and $\theta + \Delta\theta$ with the same fixed degree of freedom for all atoms of the molecule m as in the Cartesian case. This would be possible for one atom, but a second atom would be moved in a different direction due to the curvilinear coordinates. The molecule would then no longer retain its shape. Therefore, the rotation itself is still performed in the Cartesian space, but the axes of rotation are defined as in spherical coordinates.

The angle ϕ rotates the molecule around the z axis. θ rotates the molecule around the axis, which is given by the angle ϕ in the $\overline{\mathbf{xy}}$ plane. In the case that the molecular orientation happens to be exactly along the z axis, there is the problem that the angle ϕ leaves the molecule invariant. Since the molecule is linear and thus invariant under a rotation about the molecule long axis, the polar representation, as used here, results in another unintentional reduction of the degrees of freedom for the rotation. To prevent this behavior, simply the z-axis is swapped with the x-axis in this case. Then the rotation principle is retained and no further reduction of the degrees of freedoms can occur.

For the BFGS algorithm, however, the positioning formula changes again, since other rotation matrices are used for the rotation description:

$$\mathbf{r}_{i\in m} = \hat{R}_{\mathbf{n}_{\theta_m}}(\theta_m)\hat{R}_{\mathbf{n}_{\phi_m}}(\phi_m)\left[\tilde{\mathbf{r}}_{m_i}\right] + \mathbf{M}_m + \mathbf{T}_m(T_x, T_y, T_z)\ . \tag{2.7.11}$$

This also changes the rotational gradients and the vibrational gradients according to the derivation of the position formula. The rotation matrices are now formed from the general rotation matrix $R_{\mathbf{n}}(\omega)$ around any normed axis $\mathbf{n} = (n_x, n_y, n_z)^{\mathrm{T}}$ with the angle ω:

$$\hat{R}_{\mathbf{n}}(\omega) = \begin{pmatrix} n_x^2[1-c(\omega)] + c(\omega) & n_x n_y[1-c(\omega)] - n_z s(\omega) & n_x n_z[1-c(\omega)] + n_y s(\omega) \\ n_y n_x[1-c(\omega)] + n_z s(\omega) & n_y^2[1-c(\omega)] + c(\omega) & n_y n_z[1-c(\omega)] - n_x s(\omega) \\ n_z n_x[1-c(\omega)] - n_y s(\omega) & n_z n_y[1-c(\omega)] + n_x s(\omega) & n_z^2[1-c(\omega)] + c(\omega) \end{pmatrix}\ . \tag{2.7.12}$$

Again, the s and c terms are shortcuts for the sine and cosine, followed by the angle as input. For $\hat{R}(\phi_m)$, the axis of rotation $\mathbf{n}_{\phi_\mathbf{m}}$ is the z axis and for $\hat{R}(\theta_m)$ the axis

is rotated $\mathbf{n}_{\theta_\mathbf{m}}$ on a circle around the z axis, depending on ϕ_m:

$$\mathbf{n}_{\phi_m} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \mathbf{n}_{\theta_m} = \begin{pmatrix} -\sin(\phi_m) \\ \cos(\phi_m) \\ 0 \end{pmatrix} . \tag{2.7.13}$$

And in case the molecule is parallel to the z axis, the axes are simply reversed. In this case the definition of the axes change to:

$$\mathbf{n}_{\phi_m} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{n}_{\theta_m} = \begin{pmatrix} 0 \\ -\sin(\phi_m) \\ \cos(\phi_m) \end{pmatrix} . \tag{2.7.14}$$

The angles ϕ_m and θ_m remain the degrees of freedom for the molecule m and must not be specified here. The starting values of theses variables are again defined as zero. The gradients are now derived again from the position formula. Mainly thereby sine and cosine terms are exchanged, since only the rotation matrices change.

Theoretically, the shape of a molecule may change during relaxation. Imagine a single linear molecule in vacuum. This is completely relaxed and wonderfully linear. But then it is put into a crystal with other linear molecules whose tips always point to the side of another molecule. Due to the presence of the other molecules, the linear molecule is deformed. It remains no longer linear. For the algorithm, this means that the coordinate transformation into the molecular system does not work properly, because in the middle of optimization a vibration state can change to a rotation state and vice versa. On the other hand, a coordinate transformation is done only. All $3N_m$ degrees of freedom of the molecule are still considered, which means that each atom can still move independently from the other atoms in each direction. In the case that a rotational degree of freedom is lacking, this can be compensated with skillful choice of vibration conditions and vice versa. Thus, in such cases the description with the starting basis may not be the best choice but definitely not a wrong choice. In the worst case, the relaxation takes longer as if the calculation is canceled in order to create a new basis. But there may also be cases, for example, because of the Muffin-tin errors, where it is no longer possible to find the minimum along the more complicated path. In such cases, only aborting the calculation and redetermining the basis for the coordinate transformation will help.

Within a calculation, the basis of the coordinate transformation must not change. This would result in an exchange of degrees of freedom and this is not possible in an ongoing BFGS calculation.

But these are very special cases. Most molecules are not linear and even for the linear molecules the description is in the most cases stable enough.

2.8 Summary of the Algorithm

The energy of a crystal structure should be minimized. The BFGS algorithm is used for this. This algorithm iteratively proposes new atomic positions leading to a lower energy by using the positions of the atoms and the gradient of the energy. These atomic positions depend only on the initial position and the degrees of freedom of the molecules:

$$\mathbf{r}_{i\in m} = \hat{R}\left(\mathbf{r}_i^{k=0} - \mathbf{M}_m + \sum_\mu v_\mu \mathbf{V}_\mu\right) + \mathbf{M}_m + \mathbf{T}_m \ . \tag{2.8.1}$$

The rotation matrices $\hat{R}$ depend on how the molecule is constructed. In general, it is the 3 simple Cartesian rotation matrices around the respective coordinate axes. However, if the molecule is linear, then two rotation matrices are sufficient, since the molecule also loses rotational freedom. The vibration three-vectors $\mathbf{V}_\mu$ are obtained from the basis transformation of the Cartesian coordinates to molecular coordinates. In addition to the parameterized degrees of freedom f_ν, the algorithm requires the gradients for these degrees of freedom:

$$\frac{\partial E}{\partial f_\nu} = \sum_{i\in m} \frac{\partial E}{\partial \mathbf{r}_i}\frac{\partial \mathbf{r}_i}{\partial f_\nu} = -\sum_{i\in m} \mathbf{F}_i \cdot \frac{\partial \mathbf{r}_i}{\partial f_\nu} \ . \tag{2.8.2}$$

These components of the gradient are obtained by a derivation of the position formula and the force $\mathbf{F}_i$ acting on the individual atom i.

Chapter 3

Results

Here, the two algorithms presented in the previous chapter are compared to the already implemented LBFGS routine. In addition, the equilibrium structure of solid anthracene is calculated at different pressures and is compared to the available experiments and theoretical calculations. Furthermore, other molecular crystals are studied, explicitly tetracene and pentacene.

There are many optimization methods that are compared in this chapter, therefore, a label notation is introduced. First, all the methods which are considered here, use the conventional BFGS algorithm. The already implemented version of this algorithm is a limited-memory algorithm and is labeled "`LBFGS`". The new implementation of this algorithm is called "`BFGS`". In these methods, all atoms can be moved independently in the three dimensional space. The rigid-rotator model, where only 3 translational and 3 rotational degrees of freedom for each molecule are taken into account, is called "`RIGID`". Finally, the method which describes the full molecular degrees of freedom by including the molecular vibrations is called "`VIB`".

The abstract BFGS algorithm is used in all cases. Thus, only the number and meaning of the degrees of freedom change. Nevertheless, it should be mentioned, that the new `BFGS` method differs from the old implementation `LBFGS` in two ways. The old one uses only a few last steps for the prediction for the next step position where the new one uses all last steps. Furthermore, the line-search may have a different implementation. Because of that, numerical results can differ for these methods.

In addition to the relaxation method, the exchange-correlation energy functional can be changed. The standard calculations are performed with the LDA functional. Calculations for additional functionals and energy corrections like the van-der-Waals correction are named by the functional or correction name.

3.1 Comparison of the Relaxation Methods

To allow for a comparison, the same system has been chosen for the different methods. The calculations were performed for the anthracene molecular crystal at a pressure of 1.1 GPa and the results can be seen in Figure 3.1.1. A crystal under pressure has been chosen because the starting configuration is the one at zero pressure. At this pressure, the structure is already relaxed so no further relaxation is needed.

For each of the new algorithms presented here, slightly different local energy minima are found. In the case of a rigid rotator model this is not surprising since the energy surface is cutted. Due to the lack of degrees of freedom, this method can not find a suitable energy minimum.

In order to compare the relaxed configurations obtained with the different methods, the distances between individual atoms in the optimized structure are evaluated. The results are listed in Table 3.1.

Table 3.1: Listing of the distances of the atoms and the optimized energies after relaxation of the anthracene molecular crystal at a pressure of 1.1 GPa, for the three relaxation methods `BFGS`, `RIGID`, and `VIB`.

Distance differences [Bohr]	`VIB vs. RIGID`	`RIGID vs. BFGS`	`BFGS vs. VIB`
average	0.0158	0.0284	0.0300
minimim	0.0045	0.0060	0.0086
maximum	0.0336	0.0909	0.0916
standart deviation	<0.0001	0.0002	0.0002
Absolute energy difference [Ha]	0.000183	0.000169	<0.0001

Visually, nevertheless, the three structures are hardly distinguishable. In all cases, the distance differences between the atoms are very small. The largest difference is found for the closest pair of atoms belonging to different molecules. The smallest difference is found between the `VIB` and `RIGID` methods.

Therefore, the new methods can be also compared with regard to the runtime. In order to be able to estimate the calculation times corresponding to each method, the number of called self-consistent (SCF) calculations may be used. This differs from the number of optimization steps as each step goes through a number of function calls through the line-search. In general, however, the average line search rate is fairly constant, so that the number of computation steps gives a good estimate for comparing the run times.

`BFGS` required 1325 SCF loops, 427 SCF loops are used by `RIGID`, and 422 SCF loops are needed for the full molecular description `VIB`. Even if the SCF loop number

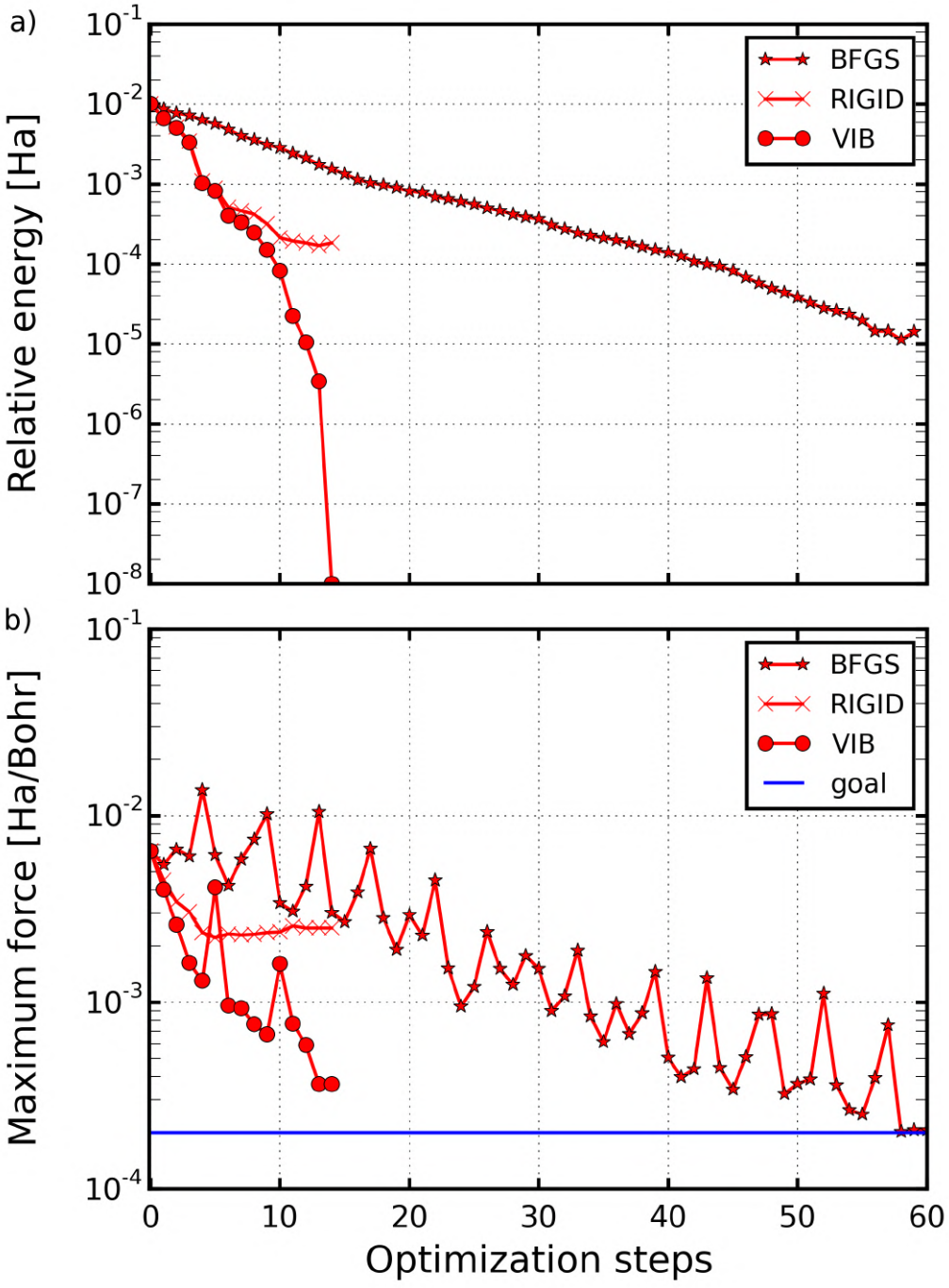

Figure 3.1.1: Comparison of three relaxation methods: `BFGS`, `RIGID`, and `VIB`. The LDA functional is used for all these calculations. The molecular crystal under consideration is anthracene at a pressure of 1.1 GPa. The starting configuration is the relaxed structure at zero pressure. a) The evolution of the energy during the BFGS-steps. The reference value for all relative energies is the smallest measured energy delivered by the vibration-enhanced method (`VIB`). b) The maximum force component acting on the atoms during the optimization steps.

is not directly related to the computational speed, for a ground-state calculation on average, this number is quite constant. The real computational times are around 89k seconds for the `BFGS` method, 30 ksec for the `RIGID` method and 30 ksec for the `VIB` method.

Both molecular models (`RIGID` and `VIB`) require the same number of optimization steps until converged, and they required almost the same SCF loops and computational time. So these two methods are close to equally fast in this case and significantly faster than the conventional description.

The disadvantage of the molecular methods, however, is that they do not relax the forces sufficiently to zero. This is not surprising for the rigid rotator. In this model, the degrees of freedom corresponding to a displacement of the atoms within a molecule are missing. Therefore, the internal distance of two atoms in the same molecule is always the same. Since the force on one atom is mostly from neighboring atoms, this force component naturally remains unchanged. With the `RIGID` model, only the force components that exist between the molecules can be reduced. So the molecules can be aligned with each other, but stresses within the molecule are retained.

In order to take the internal stresses into account in relaxation, the vibration model has been added. Thus, all degrees of freedom of the molecule can be described and influence the internal force components. Nevertheless, this algorithm does not reach the default standard force target of 0.0002 Ha/Bohr. To understand this, a closer look at the termination criteria of the BFGS algorithm is required. The BFGS algorithm terminates when either the predetermined accuracy of the gradients is reached, the energy no longer changes after an optimization step, or a maximum number of computational steps is reached. Ideally, the gradient termination condition should be taken. When the maximum number of steps has been reached, in most cases this means that the start input was too bad and ill-conditioned. And if the energy does not change after an optimization step, even though other coordinates have been chosen, then the position updates are so small that they go under in the noise of the SCF loop. However, this should only happen if the gradients are much too small to suggest such small position updates.

After the sufficient condition of a minimum, the gradient must vanish at the minimum or close to zero. For the independent atomic method `BFGS`, the gradients are exactly the force in the respective direction. Thus, at the energy minimum also the force components will disappear.

For the complete description of the molecule, however, the gradients are no longer the force components alone but according to the gradient calculation in Eq. (2.6.9) the sum of the force component multiplied by a kind of Jacobian matrix

of the coordinate transformation as shown in appendix A.3. Although the gradient is still aiming towards zero, this does not mean that the individual components, especially the force components, strive evenly towards zero.

To reduce the force further, the mathematical limitation of about 10^{-3} Ha/Bohr have to be investigated in more detail. Or a simple conventional relaxation must be done after the molecular relaxation. This should not take as long, because the structure should be very close to the desired minimum.

3.2 Anthracenes under Pressure

The determination of the angles of the anthracene molecular crystal has been performed using the new relaxation method and different exchange-correlation functionals. In order to be able to compare the calculations, the same starting atomic positions has been chosen for all different systems. For the starting positions, the relaxed position of `LBFGS` with LDA at zero pressure is taken.

Several series of optimization have been started. In each series of optimization, the relaxation method and the exchange-correlation functionals have been kept the same and only the unit cell is changed according to the given pressure.

The first series of calculations have been done using the independent-atoms methods `LBFGS` and `BFGS`. Then, the methods `RIGID` and `VIB` have been employed. For the sake of comparison among the methods and the sources, the LDA functional is used for all these calculations. The reference values (`REF`) have been calculated in the Ref. [24] and the experimental values (`EXP`) are from Ref. [26]. All LDA calculation series start with 1.1 GPa pressure because the zero pressure is already relaxed from start with the LDA functional. The angles are determined using the evaluation script again described in Section 2.5.

All simulation series provide similar results at higher pressure. The angles θ and χ were systematically overestimated and the angle δ was a little underestimated. However, the theoretical values do not fluctuate as much as the experimental ones. Thus it can be seen that most of the measurement series are still within the error range, defined with this fluctuation. The experimental δ and θ values in particular vary over most of the calculation series. This means that the respective minimum found are essentially close to the desired global minimum, as the experimental values dictates. At lower pressure and especially at zero pressure, however, some differences are visible. For instance, the θ values could not drop as fast as the experimental values.

In addition to the calculation series with the LDA functional, the complete series

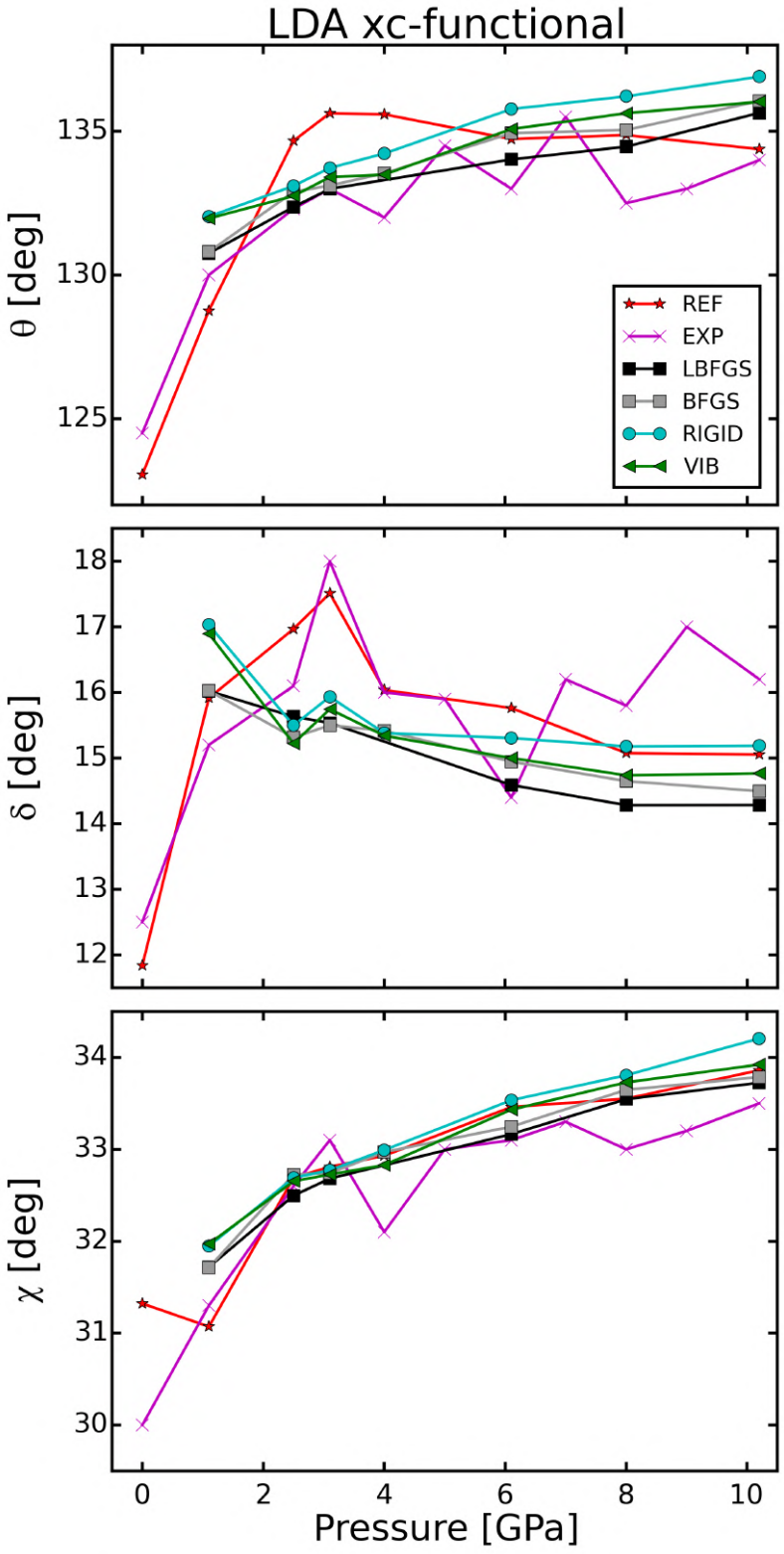

Figure 3.2.1: Dependence of the angles θ, δ, and χ on the pressure for the relaxed anthracene molecular crystal. Stars and crosses (red tint) denote reference and experimental values, respectively [24, 26]. The filled points correspond to the present series of calculations. For all series, the same starting configuration of the system and the same functional LDA was used, only the method of energy minimization was replaced. The corresponding numerical values of the angles are listed in Table 3.2.

Table 3.2: The angles θ, δ, and χ of the relaxed anthracene molecular crystal at different pressures and for different relaxation methods, as shown in figure 3.2.1. All calculation used the LDA exchange-correlation functional.

pressure [GPa]	0	1.1	2.5	3.1	4.0	6.1	8.0	10.2
θ BFGS	129.81	130.81	132.89	133.11	133.54	134.93	135.05	136.05
δ BFGS	17.96	16.03	15.32	15.50	15.42	14.95	14.65	14.50
χ BFGS	32.19	31.71	32.72	32.74	32.97	33.25	33.65	33.79
θ RIGID	129.81	132.03	133.10	133.72	134.23	135.78	136.21	136.90
δ RIGID	17.96	17.03	15.50	15.94	15.38	15.31	15.178	15.19
χ RIGID	32.19	31.95	32.69	32.77	32.99	33.53	33.81	34.21
θ VIB	129.81	131.97	132.75	133.40	133.49	135.08	135.63	136.03
δ VIB	17.96	16.90	15.22	15.75	15.34	15.00	14.74	14.77
χ VIB	32.19	31.97	32.65	32.73	32.83	33.43	33.73	33.92
θ DFTD2	131.71	132.44	133.90	134.67	134.76	135.88	136.24	136.49
δ DFTD2	4.08	16.01	15.51	16.10	15.71	15.62	15.55	15.52
χ DFTD2	39.57	33.31	33.16	33.17	33.23	33.74	33.99	34.16
θ TSvdW	129.71	131.74	132.94	133.29	133.50	135.36	135.74	136.10
δ TSvdW	20.04	15.66	14.22	14.45	14.13	14.02	14.05	14.05
χ TSvdW	32.29	32.13	32.80	32.84	32.95	33.50	33.67	33.85

is performed using the GGA functional. With different functionals, the energy on the starting configuration differs slightly and , therefore, a relaxation can be computed again in this point. The results for the GGA functional are shown in Figure 3.2.2. In this figure, the angles are obtained including two approaches to the van-der-Waals interaction, the `DFTD2` and `TSvdW`.

Table 3.3: The angles of the relaxed anthracene molecular crystal at different pressures and for different relaxation methods, as plotted in figure 3.2.2. All series here used the GGA exchange-correlation functional.

pressure [GPa]	0	1.1	2.5	3.1	4.0	6.1	8.0	10.2
θ `RIGID`	129.02	132.70	133.76	134.29	134.57	136.16	136.69	137.14
δ `RIGID`	14.74	16.26	15.59	16.00	15.65	15.35	15.25	15.23
χ `RIGID`	30.47	31.53	32.37	32.57	32.75	33.37	33.76	34.048
θ `VIB`	129.16	132.64	133.66	134.12	134.24	135.62	136.20	136.40
δ `VIB`	15.05	16.30	15.57	15.94	15.57	15.26	15.01	14.94
χ `VIB`	30.51	31.64	32.41	32.60	32.72	33.34	33.69	33.85
θ `DFTD2`	129.51	133.23	134.48	135.02	135.21	136.20	136.42	136.74
δ `DFTD2`	26.70	16.11	15.70	16.23	16.04	15.84	15.75	15.69
χ `DFTD2`	37.64	32.17	32.83	32.88	33.16	33.61	33.87	34.07
θ `TSvdW`	130.11	132.45	133.65	133.94	134.49	135.91	136.18	136.53
δ `TSvdW`	17.98	15.85	14.73	14.96	14.74	14.25	14.36	14.23
χ `TSvdW`	31.40	31.84	32.61	32.67	32.89	33.36	33.65	33.79

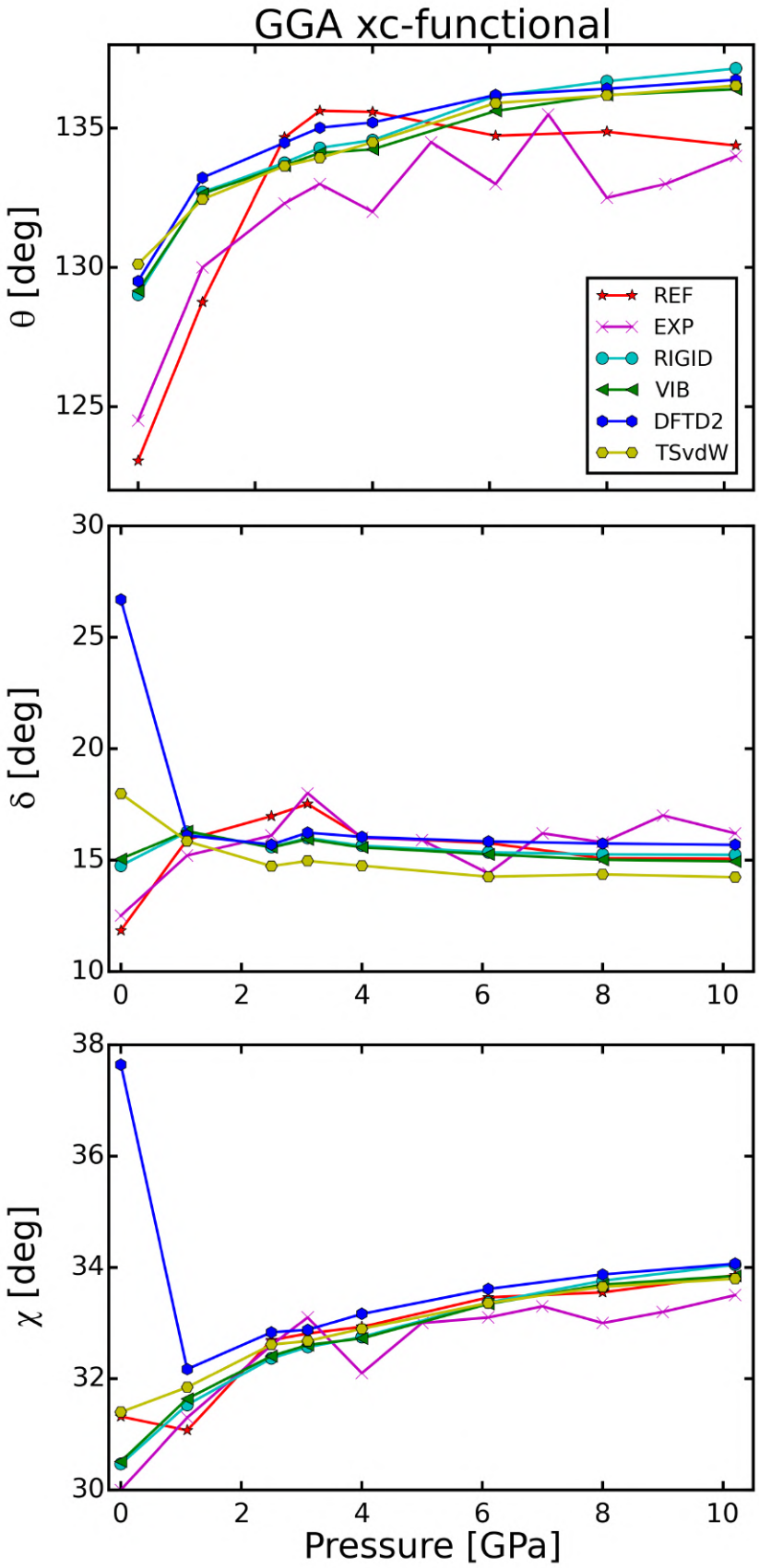

Figure 3.2.2: The angles of the different pressure series the relaxed anthracene molecular crystal compared to each other. Star and Cross denote Reference and Experimental values [24], the filled points are own series of experiments. For all series, the same starting configuration of the system and the same functional GGA was used, only the method of energy minimization was replaced, or in the case of `DFTD2` and `TSvdW`, a van-der-Waals force correction was added to the `VIB` method.

3.3 Tetracene and Pentacene

The determination of th angles is also done for the tetracene and pentacene molecular crystals. These molecules are similar to anthracene but with more hydrocarbonic rings. While the anthracene molecule has 3 rings, tetracene has four and pentacene five rings [36, 37].

Theses molecules, like anthracene, changes the values of the angels in between each other by changing the pressure. Similar test calculation series as for anthracene have been performed for tetracene and pentacene molecular crystals.

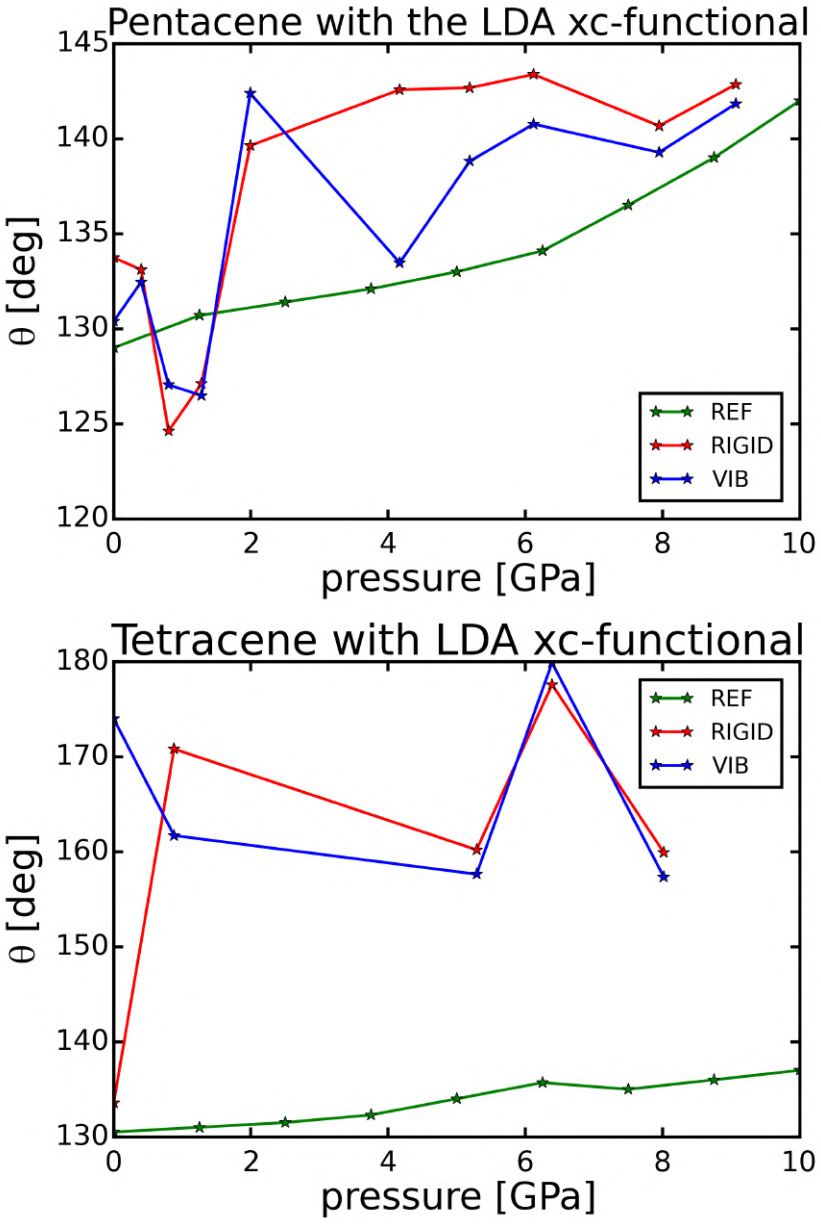

Figure 3.3.1: The angle θ of the optimized tetracene and pentacene molecular crystal relaxed with the methods `RIGID` and `VIB`. The refference values are taken from Ref. [36] and the lattice vector parameters for the pressured structure comes from Ref. [37].

Unlike the anthracene crystal, tetracene has a pressure induced phase transition at about 6 GPa. This behavoir was problematic for the test series because of the fact that the algorithm only determine local energy minima. In order to compare the results of the different methods, the starting configuration was the same for all calculation of a series. Always, the calculation starts from the configuration at zero pressure. This initial configuration was not able to determine the angles well enough. For all calculated structures, the algorithms found a local minimum. But the found minimum was not the desired global minimum as found in nature. So the algorithm works well but the starting configuration was ill-conditioned.

Monte Carlo simulations for tetracene under pressure are present in the literature Ref. [38]. In the cited work, it is shown tetracene has 268 distinct minima in eight different space groups. For the minimum energy search, this means, that there are many local minima to find, even if the introduction of pressure change the energy surface. So the found minimum has not to be necessarily the one with the lowest energy.

For the calculation this means, the desired local minimum, which should be find, must be defined more closely at the initial calculation. The algorithms work well, nevertheless, they only determine local minima. In some cases, as shown for the tetracene and pentacene calculations, the procedure used in this work could not reach the global minimum. The problem is, for such a large number of local minima, that some knowledge has to be obtained before the desired configuration with the lowest energy can be found. The starting configuration has to be closer to the desired minimum.

Conclusions and outlook

Energy minimization in the crystal is a very far-reaching topic. By adjusting the degrees of freedom in order to perform the minimization, computation time could be saved. Two algorithms have been developed that accelerate the relaxation of molecules. The first reduces the degrees of freedom per molecule while regarding the molecule as rigid. And the second uses the full molecular degrees of freedoms which are obtained from the basis transformation between Cartesian coordinates and molecular coordinates. These algorithms have been tested on the anthracene molecular crystal in order to search the angles between the molecules. For anthracene, several pressure series were calculated to compare the results of the presented methods. The optimization series have been done with the DFT exchange-correlation functionals LDA and GGA. In addition, some series with a van-der-Waals correction have been performed. The results match the results from experimental values.

However, the methods presented here, should to be tested on a larger number of molecular crystals and, eventually, using more exchange-correlation functionals. In addition to this, if too many local minima exist, then, *e.g.*, a Monte Carlo simulation should be made.

Furthermore, the lattice parameters of the unit cell a,b,c,α,β, and γ are always fixed here and are based on experimental values. For a complete structure optimization, these parameters should be added to the degrees of freedoms of the atoms or molecules. For this reason, a new method should be developed, where gradients of the lattice parameters have to be included.

It should be investigated whether the mathematical limitation of the force reduction by the Jacobian matrix can be circumvented in the vibrational method. Furthermore, an advanced BFGS method would be nice, which can also collect in a line search step information about the function to be minimized. At the moment, the information of the very computational-intensive function calls of the line search is only used for step-size control. The information included in these function calls may be also be used to improve the quality of the Hessian matrix.

Bibliography

[1] Andreas Öchsner, Holm A.: *Properties and Characterization of Modern Materials*. http://dx.doi.org/10.1007/978-981-10-1602-8. http://dx.doi.org/10.1007/978-981-10-1602-8

[2] Czycholl, Gerd: *Theoretische Festkörperphysik: Von den klassischen Modellen zu modernen Forschungsthemen*. Springer-Verlag Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-540-74790-1. http://dx.doi.org/10.1007/978-3-540-74790-1

[3] https://www.nomad-coe.eu/

[4] Walter Borchardt-Ott, Heidrun S.: *Kristallographie, Eine Einführung für Studierende der Naturwissenschaften*. Springer Spektrum. http://dx.doi.org/10.1007/978-3-662-56816-3. http://dx.doi.org/10.1007/978-3-662-56816-3

[5] D. Sauermann, H.-D. B.: *Chemie für Quereinsteiger Band 6, ristallstrukturen mit zusammengesetzten Gitterbausteinen*. Schüling

[6] Jorge Nocedal, Stephen J. W.: *Numerical Optimization*. Springer, 2000. – ISBN-13: 978-0387-30303-1

[7] Liu, Jun S.: *Monte Carlo Strategies in Scientific Computing*. Springer. http://dx.doi.org/10.1007/978-0-387-76371-2. http://dx.doi.org/10.1007/978-0-387-76371-2. – ISBN 978-0-387-76371-2

[8] Armijo, Larry: Minimization of functions having Lipschitz continuous first partial derivatives. In: *Pacific J. Math.* 16 (1966), Nr. 1, 1–3. https://projecteuclid.org:443/euclid.pjm/1102995080

[9] Broyden, Charles G.: The convergence of a class of double-rank minimization algorithms. In: *Journal of the Institute of Mathematics and Its Applications* 6 (1970), S. 76–90. http://dx.doi.org/10.1093/imamat/6.1.76. – DOI 10.1093/imamat/6.1.76

[10] Fletcher, Roger: A New Approach to Variable Metric Algorithms. In: *Computer Journal* 13 (1970), Nr. 3, S. 317–322. http://dx.doi.org/10.1093/comjnl/13.3.317. – DOI 10.1093/comjnl/13.3.317

[11] Goldfarb, Donald: A Family of Variable Metric Updates Derived by Variational Means. In: *Mathematics of Computation* 24 (1970), Nr. 109, S. 23–26. http://dx.doi.org/10.1090/S0025-5718-1970-0258249-6. – DOI 10.1090/S0025–5718–1970–0258249–6

[12] Shanno, David F.: Conditioning of quasi-Newton methods for function minimization. In: *Mathematics of Computation* 24 (1970), Nr. 111, S. 647–656. http://dx.doi.org/10.1090/S0025-5718-1970-0274029-X. – DOI 10.1090/S0025–5718–1970–0274029–X

[13] Hohenberg, P. ; Kohn, W.: Inhomogeneous Electron Gas. In: *Physical Review* 136.3B (Nov. 1964), Nr. B864-B871. http://dx.doi.org/10.1103/PhysRev.136.B864. – DOI 10.1103/PhysRev.136.B864

[14] Tsuneda, Takao: *Density functional theory in quantum chemistry.* Tokyo : Springer, 2014 http://cds.cern.ch/record/2278901

[15] Sham, L. J. ; Kohn, W.: One-Particle Properties of an Inhomogeneous Interacting Electron Gas. In: *Phys. Rev.* 145 (1966), May, 561–567. http://dx.doi.org/10.1103/PhysRev.145.561. – DOI 10.1103/PhysRev.145.561

[16] Perdew, John P. ; Chevary, J. A. ; Vosko, S. H. ; Jackson, Koblar A. ; Pederson, Mark R. ; Singh, D. J. ; Fiolhais, Carlos: Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. In: *Phys. Rev. B* 46 (1992), Sep, 6671–6687. http://dx.doi.org/10.1103/PhysRevB.46.6671. – DOI 10.1103/PhysRevB.46.6671

[17] Ortiz, G. ; Ballone, P.: Pseudopotentials for non-local-density functionals. In: *Phys. Rev. B* 43 (1991), Mar, 6376–6387. http://dx.doi.org/10.1103/PhysRevB.43.6376. – DOI 10.1103/PhysRevB.43.6376

[18] Hammer, B. ; Hansen, L. B. ; Nørskov, J. K.: Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals. In: *Phys. Rev. B* 59 (1999), Mar, 7413–7421. http://dx.doi.org/10.1103/PhysRevB.59.7413. – DOI 10.1103/PhysRevB.59.7413

[19] Perdew, John P. ; Burke, Kieron ; Ernzerhof, Matthias: Generalized Gradient Approximation Made Simple. In: *Phys. Rev. Lett.* 77 (1996), Oct, 3865–3868. http://dx.doi.org/10.1103/PhysRevLett.77.3865. – DOI 10.1103/PhysRevLett.77.3865

[20] Grimme, Stefan: Semiempirical GGA-type density functional constructed with a long-range dispersion correction. In: *Journal of Computational Chemistry* 27.15 (2006)

[21] Tkatchenko, Alexandre ; Scheffler, Matthias: Accurate molecular van-der-Waals interactions from ground-state electron density and free-atom reference data. In: *Phys. Rev. Lett.* 102 (2009), Nr. 073005

[22] Gulans, Andris ; Kontur, Stefan ; Meisenbichler, Christian ; Nabok, Dmitrii ; Pavone, Pasquale ; Rigamonti, Santiago ; Sagmeister, Stephan ; Werner, Ute ; Draxl, Claudia: exciting: a full-potential all-electron package implementing density-functional theory and many-body perturbation theory. In: *Journal of Physics: Condensed Matter* 26 (2014), Nr. 36, 363202. http://stacks.iop.org/0953-8984/26/i=36/a=363202

[23] http://exciting-code.org

[24] Kerstin Hummer, Peter P. ; Ambrosch-Draxl, Claudia: Ab initio study of anthracene under high pressure. In: *DOI: 10.1103/PhysRevB.67.184105* (2003)

[25] Mason, R.: The crystallography of anthracene at 95°K and 290°K. In: *Acta Crystallographica* 17 (1964), May, Nr. 5, 547–555. http://dx.doi.org/10.1107/S0365110X64001281. – DOI 10.1107/S0365110X64001281

[26] M. Oehzelt, A. N. R.Resel: High-pressure structural properties of anthracene up to 10 GPa. In: *Phys. Rev.* B (2002), Nr. 66. http://dx.doi.org/https://doi.org/10.1103/PhysRevB.66.174104. – DOI https://doi.org/10.1103/PhysRevB.66.174104

[27] Zhao, Liang ; Baer, Bruce J. ; Chronister, Eric L.: High-Pressure Raman Study of Anthracene. In: *The Journal of Physical Chemistry A* 103 (1999), Nr. 12, 1728-1733. http://dx.doi.org/10.1021/jp9838994. – DOI 10.1021/jp9838994

[28] http://www.scipy-lectures.org/advanced/mathematical_optimization/

[29] T. Betts, John ; Campbell, Stephen: Discretize then optimize. (2005), 01

[30] Betts, John T.: *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming.* 2nd. New York, NY, USA : Cambridge University Press, 2009. – ISBN 0898716888, 9780898716887

[31] Sundius, Tom: *Molvilb*, June 2002. www.mv.helsinki.fi/home/sundius/MOLVIB/molv7man.pdf. – Helsinki

[32] Joseph W. Ochterski, Ph.D.: *Vibrational Analysis in Gaussian.* 1999. – URL: http://gaussian.com/vib/ Mail: help@gaussian.com

[33] Andrzej Kielbasiński, Hubert S.: *Numerische lineare Algebra. Eine computerorientierte Einführung.* Deutscher Verlag der Wissenschaften, Berlin 1988. – ISBN 3-326-00194-0

[34] Hermann, Martin: *Numerische Mathematik.* Oldenbourg Verlag, München und Wien 2006. – ISBN 3-486-57935-5

[35] Arnoldi, W.E.: The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem. In: *Quarterly of Applied Mathematics* (1951), S. 17–29

[36] Schatschneider, Bohdan ; Monaco, Stephen ; Tkatchenko, Alexandre ; Liang, Jian-Jie: Understanding the Structure and Electronic Properties of Molecular Crystals Under Pressure: Application of Dispersion Corrected DFT to Oligoacenes. In: *The Journal of Physical Chemistry A* 117 (2013), Nr. 34, 8323-8331. http://dx.doi.org/10.1021/jp406573n. – DOI 10.1021/jp406573n. – PMID: 23901832

[37] Oehzelt, M. ; Aichholzer, A. ; Resel, R. ; Heimel, G. ; Venuti, E. ; Della Valle, R. G.: Crystal structure of oligoacenes under high pressure. In: *Phys. Rev. B* 74 (2006), Sep, 104103. http://dx.doi.org/10.1103/PhysRevB.74.104103. – DOI 10.1103/PhysRevB.74.104103

[38] Della Valle, Raffaele G. ; Venuti, Elisabetta ; Brillante, Aldo ; Girlando, Alberto: Inherent Structures of Crystalline Tetracene. In: *The Journal of Physical Chemistry A* 110 (2006), Nr. 37, 10858-10862. http://dx.doi.org/10.1021/jp0611020. – DOI 10.1021/jp0611020. – PMID: 16970382

Appendix A

Test cases for all degrees of freedom

For such a source code to work, it must be debugged properly. So some test cases where proposed to verify, that the program works properly.

For the algorithm it is, *e.g.*, important that the positions of the atoms coincide with the degree of freedom of the BFGS algorithm and that the gradients at the respective positions are correct. The positioning can be checked relatively easily by applying the position formula and verifying that the atoms are actually at the required positions.

So an easy test case is to specify the displacement of a molecule to, *e.g.*, 3 Bohr in the x direction and rotate it 30° along the y axis. The resulting atomic positions can now be compared with the nominal position from the positioning formula.

It is more difficult to check if the gradient of one degree of freedom is correct. As in chapter 2.4 shown, depending on the method used and the approximation made to determine the gradients, they may differ slightly. In addition, it must be ensured that the discretization has a sufficiently small influence on the gradients. To check if the gradients used here have been chosen correctly, the following formula derived from the differential quotient can be used:

$$f(\vec{X} + \vec{h}) - f(\vec{X}) - \vec{h} \cdot \vec{\partial} f(\vec{X}) = \mathcal{O}(\vec{h}^2) \tag{A.1}$$

Here the parametrization $\vec{h} = \epsilon \vec{k}$ with the factor ϵ will be used in order to determine the behavior of the function values when letting ϵ go to zero. $\vec{k}$ is a kind of search direction as in the BFGS line search and can be chosen arbitrarily. $\vec{X}$ is the local point where the gradient behavior along the line $\vec{k}$ is checked.

If the values $f(\vec{X} + \epsilon\vec{k}) - f(\vec{X}) - \epsilon\vec{k} \cdot \vec{\partial} f(\vec{X})$ are plotted to ϵ logarithmically,

then a slope of 2 has to be expected in a confidence interval. Of course, for too large values of $\vec{h}$, this is not true because too large computations of the function are skipped and the local gradient $\vec{\partial} f(\vec{X})$ is just too far away. For too small increments of $\vec{h}$, the computing inaccuracy of floating-point numbers becomes a fatal error. Too small steps generate differences from too small numbers which have high computing inaccuracy. In between, a slope of 2 is expected if the gradient $\vec{\partial} f(\vec{X})$ is good enough.

If $\vec{k}$ is chosen in a random direction, the convergence behavior of all gradients is obtained. But a specific degree of freedom can be chosen also while leaving all other components of $\vec{k}$ zero. In this case, the convergence behavior from the specific chosen degree of freedom is obtained. With this method, the investigation of the convergence behavior of the gradients can be done in more detail.

The vibration analysis of a molecule, as described in chapter 2.7, excluded the mass weighted coordinates which were introduced in the Gaussian source [32]. This is not the first intuitive approach for the problem. The first attempt in this work was to use the Gaussian convention $\mathbf{q}_i = \sqrt{m_i}\mathbf{r}_i$. So the translation vectors and the rotation vectors of the molecule were mass weighted. Due to the orthogonalization of the vibration vectors, the entries of these vectors are also mass weighted, too. To get a real position, the mass factor then has to be divided when applying the position formula. Finally, the position formula will also be mass weighted and therefore the derived gradients.

But with the gradient test as seen in figure A.1, it turns out that the vibration gradients are too bad. Because of that, they needed to be investigated more closely. A basis transformation was carried out for the vibration analysis in our molecules. Generally written before the transformation, the energy was evaluated at the position $\vec{X}$.

$$E = E(\vec{X}) \qquad \vec{\partial} E = -\vec{F} \tag{A.2}$$

Then the basis transformation $T(\vec{Y}) = \vec{X}$ was introduced, which changes the energy calculation.

$$\tilde{E} = \tilde{E}(\vec{Y}) = E(T(\vec{Y})) \qquad \vec{\partial}\tilde{E} = J(\vec{Y}) \cdot \partial E(T(\vec{X})) \tag{A.3}$$

Now, if the energy is to be minimized, then it is known that at least the gradients become exactly zero or become smaller and smaller near the minimum. In the case without a coordinate transformation, this means that the forces acting on the atoms strives towards zero. In the case of the coordinate transformation, the gradient still goes to zero, but this is now composed of the force and a kind of Jacobian matrix $J(\vec{Y})$ of the transformation. The product goes to zero, but that does not mean the force has to go to zero. If the Jacobian matrix is printed out, then it is notable that the vibration entries are very small. This results in a small gradient for the

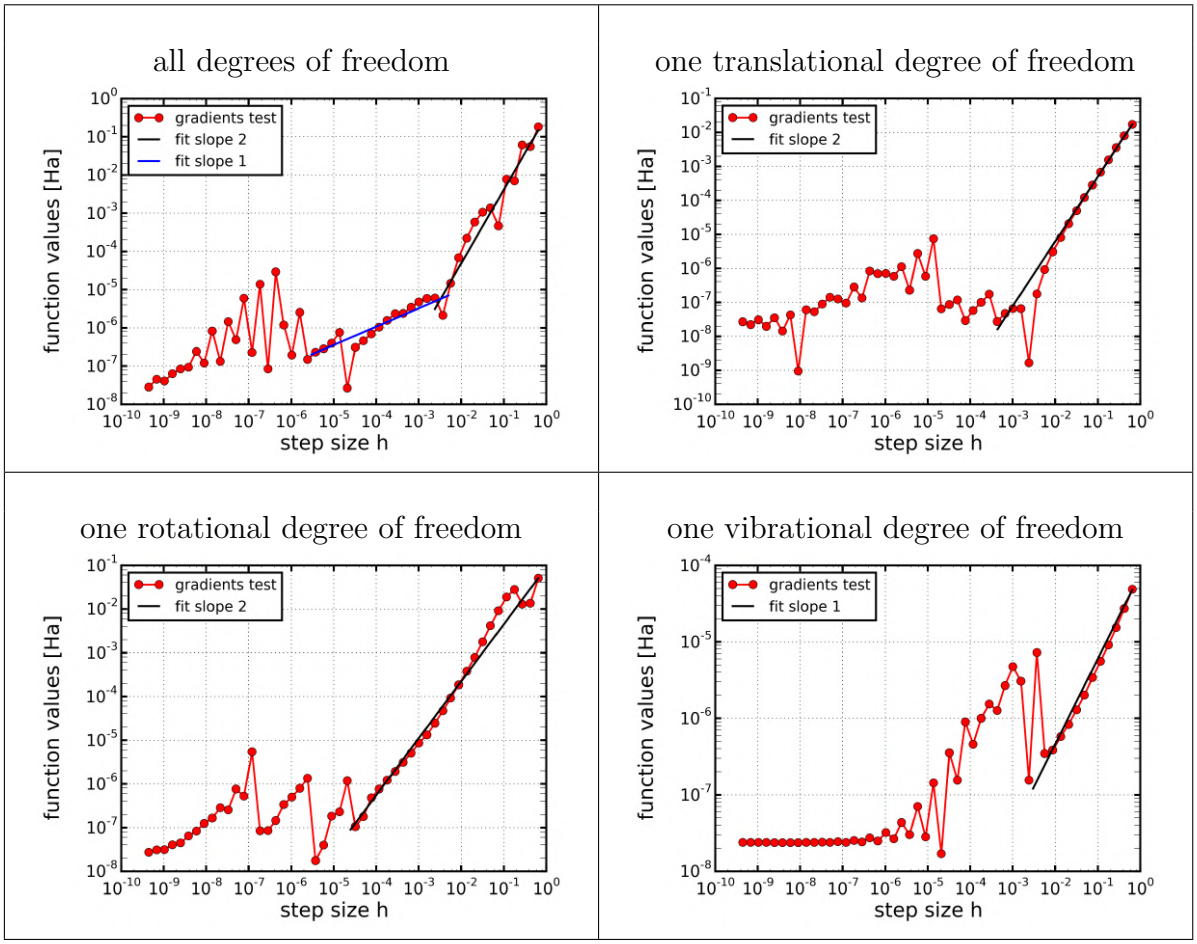

Figure A.1: Gradient tests at the starting position of all Anthracene calculations. The behavior of the nearby gradients (distance $\vec{h}$ from the random direction $\vec{k}$) of the starting point is compared with the gradient at the starting point. The error should be quadratic according to the formula A.1, so the expectation is a slope of 2 in the double logarithmic plot. This slope is seen up to approximately $5 \cdot 10^{-3}$. Then there is an area where the slope has only the value 1. From a step length of $5 \cdot 10^{-6}$, large computational errors occurs. This area is no longer within the confidence range. Starting with a position difference of this length, the positions are partially equal in exciting. Problems make the vibration degrees of freedom, which converge only with a slope of 1 and with a smaller increment at all.

vibrational degrees of freedom. Compared to the translational and rotational degrees of freedom, the vibration degrees of freedom are about 100 times smaller. At the same time it is visible that the vibration degrees of freedom can be in the order of 10 to 100. In comparison, *e.g.*, the rotation has already made half a turn with a value of $\pi = 3.141592$, or translation has traversed a complete unit cell at about 14 for the anthracene crystal. Then the vibrational degrees of freedom are very large and the effect of their movement is weak.

For the BFGS algorithm, it is basically not important which magnitude the gradient has. The decisive factor is the direction of the gradient. The length of the search in this direction is indeed induced by the magnitude of the gradients but with sufficient BFGS steps in advance, the Hessian matrix is set well enough to set the length properly. And for a more accurate search near the suggested length, there is still the line-search. Nevertheless, the different gradients magnitudes pose a problem, since there is a multi-level relaxation. First, the larger gradients are reduced and only if this is sufficiently done, the smaller gradients could disappear.

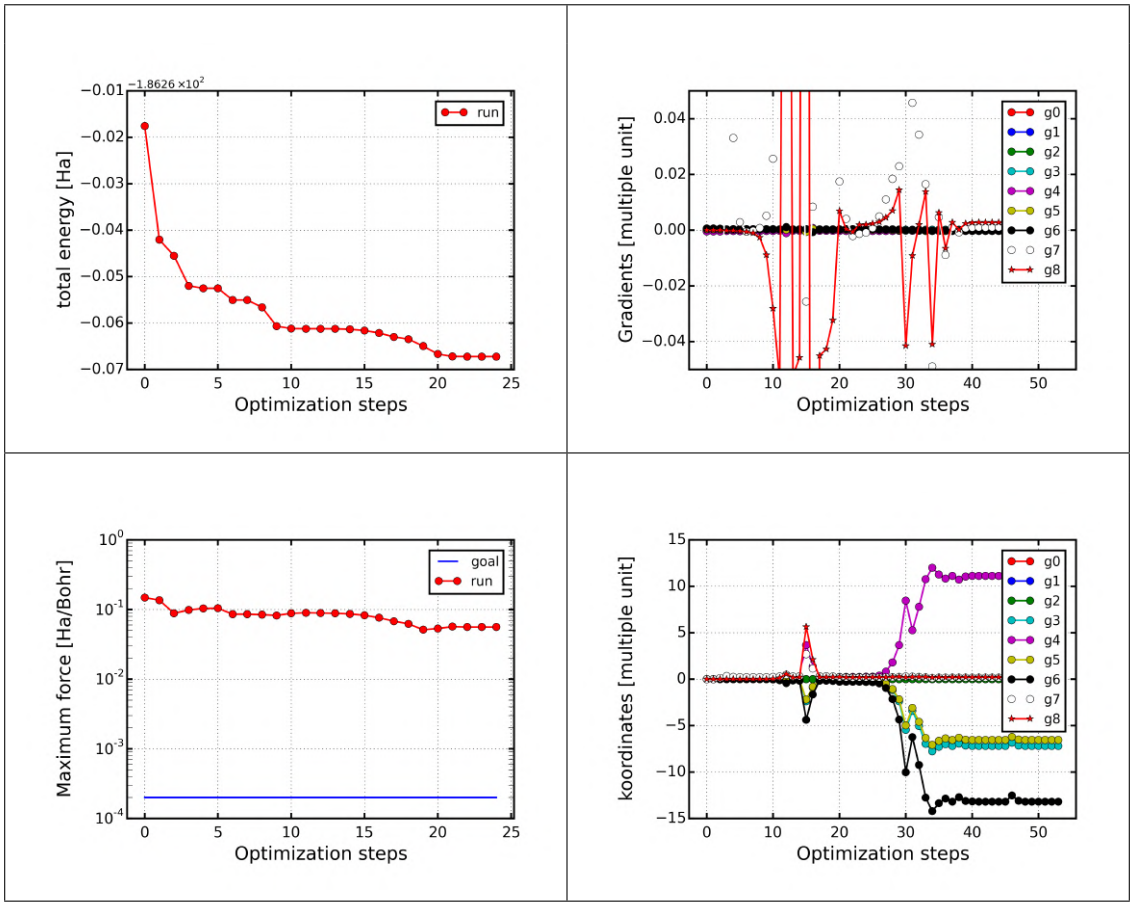

Figure A.2: Relaxation of a single CO_2 molecule. The vibration basis is mass-weighted, resulting in big degrees of freedom. The energy is gradually reduced and the maximum forces are not significantly reduced. The energy and force is displayed for each BFGS step, while the gradients and degrees of freedom (koordinates) are also shown for each line-search step.

The cause lies in the normalization of the vibration states in the context of a very large mass term. By normalizing, the magnitude of the basis is adjusted relative to the translations and rotational gradients. In the creation of the vibration basis, the mass terms are taken into account, which are in the range of 20,000. After normalization, the basis has a maximum magnitude of 1, but at each vibration query it is divided by the root of the mass term, which explains the factor 100. For the vibration states, it is most useful to eliminate the mass terms as quickly as possible. At least they must not be present at the normalization. For the inertial tensor they are still needed, but even with the rotation vectors, the mass can be eliminated by doing the inverse transformation of the mass-weighted coordinates already there. Thus, all mass terms are eliminated from the positioning formula.

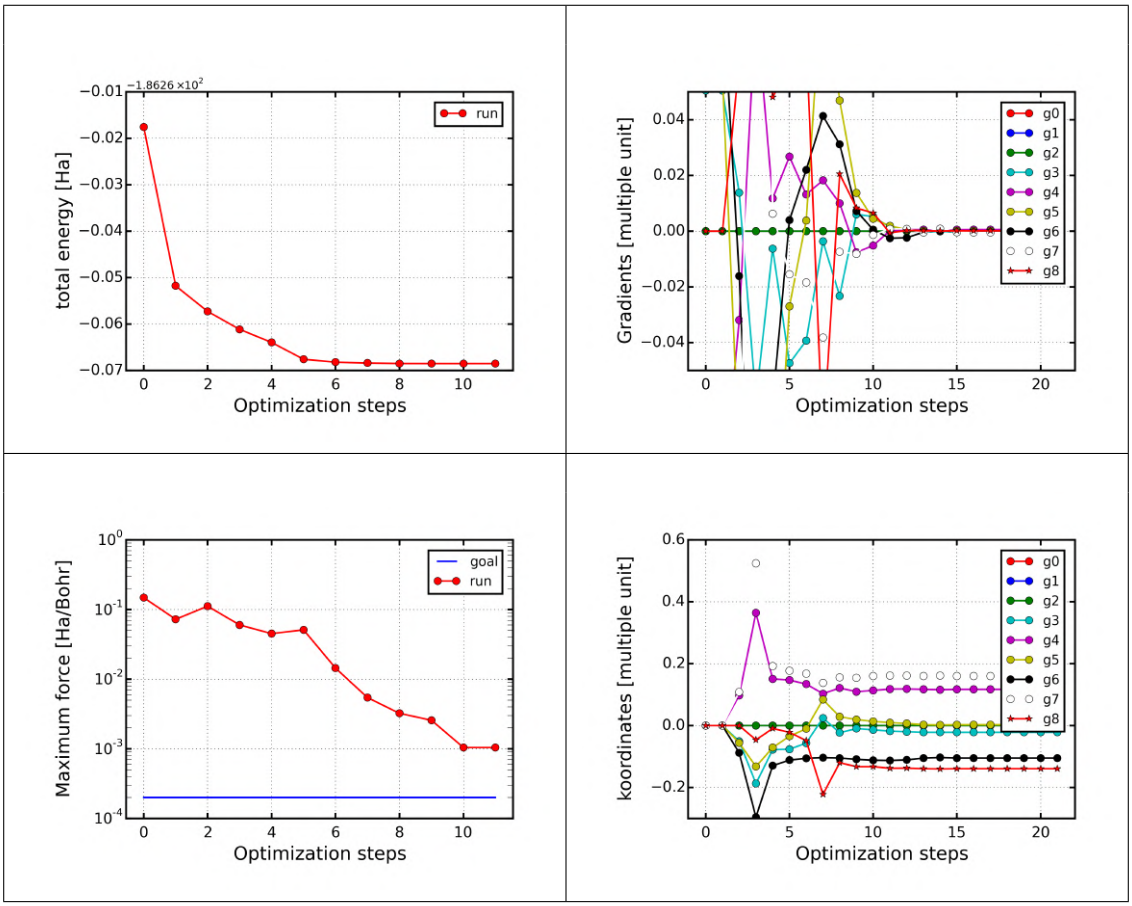

Figure A.3: Relaxation of a single CO_2 molecule. The vibration basis is no longer mass weighted. The system converges much faster in comparison with the mass-weighted results shown in Figure A.2. Also the force decreases more here, but not as much as it should. Now, the vibration degrees of freedom are of the same order of magnitude as the translation and the rotation. The energy and force is displayed for each BFGS step, while the gradients and degrees of freedom (koordinates) are also shown for each line search step.

The result is that the energy converges faster again because it no longer has the step shape. Now all gradients can equally contribute to energy minimization. But nevertheless, there remains a problem in the algorithm. This is visible by regarding

again the maximum force components. Even if the results are a way better, the force components do not reach the goal of the desired 0.0002 Ha/Bohr. In all regarded calculations, the vibration analysis leads to a lower energy than the normal BFGS-algorithm but the force components do not converged sufficiently.

Appendix B

Computational times

Since most calculations in the test series yield similar results, it is interesting to know how long these calculations took. As in section 3.1 mentioned unfortunately, there are no direct measurements. For the computation time there can not be know, which other calculations still ran on the same node in the cluster and, *e.g.*, slowed access to the memory or Hard disk space. And also the times of the SCF loops can vary depending on the problem. Both values are only indicators for how long a single configuration takes to relax with the particular method. That is why both data series are analyzed together.

Because the starting configuration of the relaxation is almost at zero pressure, this initial configuration is less suitable for increasing pressure. For the relaxation, the way in the hyperspace of the energy gets longer with increasing pressure since the initial configuration gets more and more ill-conditioned. As a result, the calculation time increases continuously, except for fluctuations. The deviations up and down arise because some configurations converge better or worse as the pressure increases.

Except for one relaxation of the `DFTD2` with the GGA functional at 6.1 GPa, all calculations converge. In this one calculation, the self-consistent loop (SCF) known from the DFT is not converged. However, the angles look reasonable and fit into the measuring series. It can be seen that the LDA calculations tend to be faster than the GGA calculations. This is not surprising since the higher order GGA also requires more computation than LDA. However, the number of computation steps required and thus the number of SCF loops depends on the number of optimization steps required. The more steps the optimization algorithm needs, the more SCF loops the program must calculate.

To compare the own methods in the computation time, the `RIGID` method is for the same functional used in most cases the fastest method. But the `Vib` method is commonly not much slower than the `RIGID`. The van-der-Waals corrections were

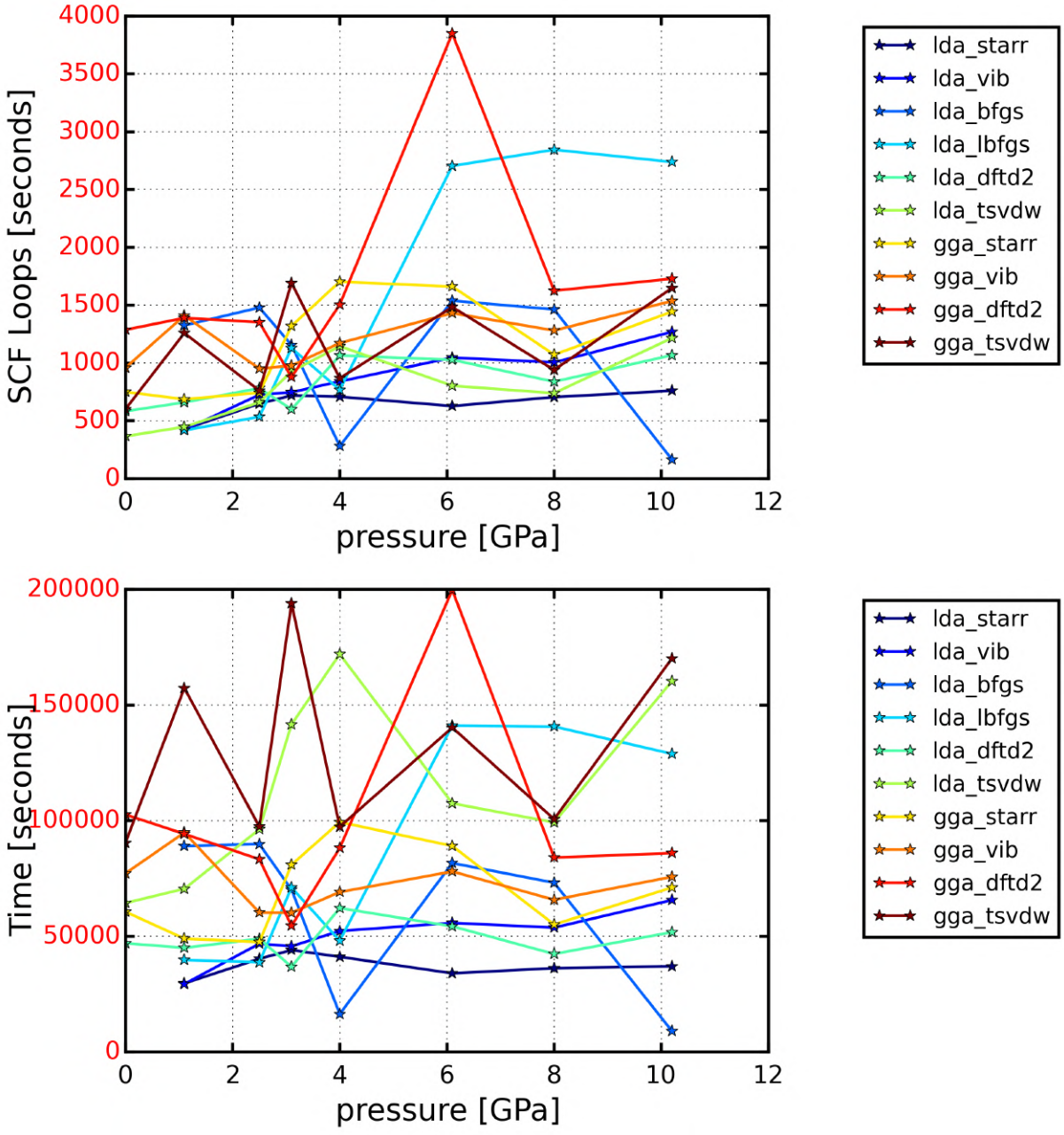

Figure B.1: The indicators for the required calculation times of the test series of anthracene. Except for one relaxation of the `DFTD2` with the GGA functional at 6.1 GPa, all calculations converge properly.

carried out with the method of vibration analysis. Compared to the test series without correction, the corrected calculations needed much more computing time. Due to the fact, that these calculations have the worst results in the angle determination, they perform badly.

The `LBFGS` method with the LDA Functional makes an exception to the calculation time. For larger pressures, this method suddenly requires significantly more optimization steps. This is because the conventional `LBFGS` method establishes an initial trust region by searching for the minimum. But if the minimum is not within this region, then the method can not find it and breaks. However, the method is written in such a way that if the structure is not yet relaxed, then the method "Harmonics" takes over. This, of course, takes a bit more computation steps than the BFGS methods and thus more function calls.

The pressure point of 4 GPa is noticeable. At this point, many optimization algorithms seem to converge very well. The BFGS method even requires only 16 optimization steps, where it is 50 steps in the case of larger and smaller pressures. With increasing pressure, this point seems to lay after a maximum δ angle, as the experimental values show in Figure 3.2.1. So may be, the relaxation from the zero pressure is a bit faster there.

Abbreviation

Operator or Symbol	Declaration
$\vec{x}$	The arrow means, that the underlying symbol, here the x, is a vector of commonly higher dimension.
$\mathbf{x}$	Bold symbols, here the x, are exact three dimensional vectors.
$\hat{B}$	The hat means, that the underlying symbol, here the B, is a matrix or an operator.
$\vec{p}^{\mathrm{T}}$	The T means the transpose of its vector or the matrix.
$\boldsymbol{\nabla}$	The nabla operator in exact three dimensions.
$\vec{\partial}$	The nabla operator in higher dimensions.
$x^{k=0}$	a degree of freedom with the power of $k = 0$ means the starting value for the zero-th iteration.

Abbreviation	Declaration
DFT	Density Functional Theory
KS	Kohn-Sham
LDA	Local Density Approximation
GGA	General Gradient Approximation
MT	muffin tin
IR	interstitial region
APW	Augmented Plane Wave
LAPW	Linear Augmented Plane Wave
LO	Local orbital functions
SCF	self-consistent loop
`LBFGS`	Method of the common BFGS-algorithm
`BFGS`	own implementation of the BFGS-algorithm
`RIGID`	rigid molecules method
`VIB`	full molecular degrees of freedom method
`DFTD2`	Van-der-Waals correction method
`TSvdW`	Van-der-Waals correction method

Symbol	Declaration	first occurrence
$f(\vec{X})$	function to be minimize	1

$\vec{X}_e$	extreme point of f	1
D	dimension of $\vec{X}$	1
k	iteration index	1
f_k	function value in the k-th iteration	1
$\vec{g}$	exact gradient of f	1.1
$\hat{B}$	exact hessian matrix of f	1
$\vec{g}_k$	approximated gradient of f in the k-th iteration	1.1
$\hat{B}_k$	approximated hessian matrix of f in the k-th iteration	1
$\vec{p}_k$	search direction for the next step k	1.1.1
A_k	length of a line-search in the k-th iteration	1.1.1
$\tilde{f}(A)$	reduced dimensionality of the function $f(\vec{X})$ along $\vec{p}_k$	1.1.1
m_k	model function of $f(\vec{X})$ in the k-th iteration	1.1.2
$\vec{s}_k$	difference between the last step positions of f_k	1.1.4
$\vec{y}_k$	difference between the last gradients of f_k	1.1.4
γ_k	factor for the rank 1 approach (Quasi-Newton)	1.1.4
$\vec{u}_k$	vector for the rank 1 approach (Quasi-Newton)	1.1.4
δ_k	factor for the rank 2 approach (BFGS)	1.1.5
$\vec{v}_k$	vector for the rank 2 approach (BFGS)	1.1.5
E_{TOT}	total energy of the many body problem	1.2
$\hat{H}$	Hamiltonian of the many body problem	1.2
Ψ	wavefunction of the many body problem	1.2
i	label for electrons (will be redefined)	1.2
I	label for atoms	1.2
$\mathbf{r}_i$	coordinate vector of the single electron numbered with i	1.2
$\mathbf{R}_I$	coordinate vector of the single nucleus numbered with I	1.2
$\vec{r}$	collection vector of the positions of all electrons	1.2
$\vec{R}$	collection vector of the positions of all nuclei	1.2
N_{nuc}	Number of nuclei in the system	1.2
N	Number of electrons in the system	1.2
$\hat{T}_{\mathrm{nuc}}$	kinetic energy operator of all nuclei	1.2
$\hat{V}_{\text{nuc-nuc}}$	nuclei interaction potential	1.2
$\hat{H}_e$	electronic part of the Hamiltonian	1.2
$\hat{T}_e$	kinetic energy operator of all electrons	1.2
$\hat{V}_{e\text{-}e}$	electronic interaction potential	1.2
$\hat{V}_{e\text{-nuc}}$	electron nuclei interaction potential	1.2
$\hbar$	Planck constant	1.2
m_e	mass of an electron	1.2
e	charge of an electron	1.2
π	mathematical constant	1.2

ϵ_0	vacuum permittivity	1.2
M_I	mass of the corresponding nucleus I	1.2
$Z_I Z_J$	charge of the corresponding nucleus I or J	1.2
$\Psi_{\mathrm{BO}}(\vec{R},\vec{r})$	wave function in the Born-Oppenheimer approximation	1.2
$\chi_{\mathrm{nuc}}(\vec{R})$	nuclei part of the wave function	1.2
$\phi_e(\vec{r},\vec{R})$	electronic part of the wave function	1.2
E_e	electronic energy	1.2
$E_{e,\mathrm{gs}}(\vec{R})$	electronic groundstate energy	1.2
N_{Atom}	total number of atoms in the unit cell	2.1
i	redefined in this chapter: label of an atom	2.1
m	label of an molecule	2.1
N_{Atom}	total number of atoms in the unit cell	2.1
x_i	x component of the position $\mathbf{r}_i$ of the atom i	2.1
y_i	y component of the position $\mathbf{r}_i$ of the atom i	2.1
z_i	z component of the position $\mathbf{r}_i$ of the atom i	2.1
$\mathbf{r}_i^{k=0}$	starting position of the atom i	2.1
$\mathbf{F}_i$	the Force acting on the atom i	2.1
N_{Mol}	total number of molecules in the unit cell	2.1
N_m	number of atoms belonging to the molecule m	2.1
$\mathbf{M}_m^{\mathrm{Geo}}$	the geometric center of the molecule m	2.6
$\mathbf{M}_m^{\mathrm{Com}}$	the center of mass of the molecule m	2.6
$\mathbf{M}_m$	center of molecule m ether geometric or mass weighted	2.6
$\mathbf{T}_m$	vector of translational degrees of freedom for molecule m	2.6
α_m	rotational degrees of freedom for molecule m around x-axes	2.6
β_m	rotational degrees of freedom for molecule m around y-axes	2.6
γ_m	rotational degrees of freedom for molecule m around z-axes	2.6
$\hat{R}_x(\alpha)$	rotation matrix for rotations of the angle α around x-axes	2.6
$\hat{R}_y(\beta)$	rotation matrix for rotations of the angle β around y-axes	2.6
$\hat{R}_z(\gamma)$	rotation matrix for rotations of the angle γ around z-axes	2.6
f_m	any molecular related degree of freedom: translation-, rotation-, or vibration coordinate	2.6
$\tilde{\mathbf{r}}_i$	molecular coordinate of the atom $i \in m$, $\mathbf{M}_m$ is the origin	2.6
$s(\omega)$	shortcut for $\sin(\omega)$	2.6
$c(\omega)$	shortcut for $\cos(\omega)$	2.6
$\mathbf{A}_i$	derivation vector $\frac{\partial \mathbf{x}_i}{\partial \alpha}$	2.6
$\mathbf{B}_i$	derivation vector $\frac{\partial \mathbf{x}_i}{\partial \beta}$	2.6
$\mathbf{C}_i$	derivation vector $\frac{\partial \mathbf{x}_i}{\partial \gamma}$	2.6
$\vec{x}_m^{\mathrm{total}}$	position vector describing the molecule m	2.7
$\vec{e}_a$	$3N_m$ dim. basis vector of the Cartesian coordinate of $\vec{x}_m^{\mathrm{total}}$	2.7

$\vec{E}_a$	$3N_m$ dim. basis vector of the molecular coordinate of $\vec{x}_m^{\text{total}}$	2.7
$\mathbf{q}_i$	mass weighted coordinates as use in gaussian [32]	2.7
$\hat{I}_m$	inertial tensor of the molecule m	2.7
$\widehat{G}_{3\times3}$	eigenvector matrix of the inertial tensor $\hat{I}_m$	2.7
ν	coordinate index for x,y and z	2.7
P_ν	scalar product of $\tilde{\mathbf{r}}_i$ and the ν-th row of $\widehat{G}$	2.7
μ	index for vibrational modes of a molecule	2.7
$\vec{V}_\mu$	$3N_m$ dimensional vibration mode μ of the molecule m	2.7
$\mathbf{V}_\mu, i$	tree-vector component of $\vec{V}_\mu$ from the atom i	2.7
ϕ and θ	coordinates in polar representation	2.7
ϕ_m	rotational degrees of freedom for molecule m	2.7
θ_m	rotational degrees of freedom for molecule m	2.7
$\hat{R}_{\mathbf{n}}(\omega)$	general rotation matrix around line n with ω	2.7

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den 31 juli 2018