UNIVERSITÄT
BAYREUTH

Excitation dynamics in molecular systems from efficient grid-based real-time density functional theory

von

Ingo Schelter

Excitation dynamics in molecular systems from efficient grid-based real-time density functional theory

Genehmigte Abhandlung
Von der Universität Bayreuth
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

von

Ingo Schelter

geboren in Marktredwitz

1. Gutachter:	Prof. Dr. Stephan Kümmel
2. Gutachter:	Prof. Dr. Matthias Ullmann
Tag der Einreichung:	19. Dezember 2016
Tag des Kolloquiums:	10. März 2016

Abstract

Making use of solar energy is a very promising approach for satisfying the energy needs of mankind. Nature has perfected its light harvesting mechanisms in plants, algae, and photosynthetic bacteria with respect to efficiency and stability. The complexes that participate in the photosynthetic process are typically aggregates of chromophores that are often anchored inside a protein scaffold. To understand the underlying processes that make natural light harvesting so efficient on the one hand and protect the participating protein-chromophore complexes from damage on the other hand is a key issue for the development of artificial light harvesting devices. Next to extensive experimental studies, this requires a detailed theoretical description on the molecular, quantum-mechanical level.

Time-dependent density functional theory (TDDFT) appears as a natural choice in this respect since it combines predictive power and reliability with computational efficiency. Especially its real-space and real-time implementation is applicable to molecular systems with hundreds or even up to a few thousands of electrons. However, utilizing the theoretical potential of TDDFT requires modern, highly parallel computers and programs that are able to exploit their full computational power. One major part of this thesis is dedicated to the development of the real-space and real-time TDDFT code BTDFT. The challenges within this part include a highly scalable parallelization and the development of data structures that allow for efficient memory access while keeping the code simple and flexible.

The reliable simulation of excitation-energy transfer (EET) within real-time TDDFT requires an accurate description of the relevant excited states and their nature. This includes excitation energies, oscillator strengths, and transition densities, which allow to visualize the spatial oscillations of the time-dependent electron density. A second part in this work addresses the accurate computation of these quantities from real-time TDDFT.

The exchange-correlation (xc) energy, which describes the non-trivial many-particle interactions within TDDFT, has to be approximated. The size of typical light-harvesting systems requires the use of lightweight approximations to exchange and correlation. These are well established and were used in the past with great success but also have well known deficiencies. The most critical one within this work is their systematic inability to describe charge-transfer processes correctly, which results from a spurious self-interaction of an electron with its own charge density. The last part within this thesis discusses the reliability of real-time TDDFT with lightweight xc approximations for EET simulations in natural systems by means of one and two aggregated chromophores that appear in nature. Their environment, which consists of a protein scaffold and other chromophores, is included into these investigations explicitly or through an electrostatic potential. Finally, the outlook is dedicated to the general real-time simulation of energy transfer processes within real-time TDDFT.

Kurzfassung

Die Verwendung der Sonnenenergie zur Deckung des stetig wachsenden Energiebedarfs der Menschheit birgt großes Potential. Die Natur hat ihre Lichtsammelkomplexe in Pflanzen, Algen und photosynthetischen Bakterien bereits im Hinblick auf Effizienz und Stabilität perfektioniert. Die an der Photosynthese beteiligten Komplexe sind typischerweise Aggregate aus Chromophoren, die in ein Proteingerüst eingebettet sind. Ein vielversprechender Ansatz zur Entwicklung effizienter Solarzellen besteht darin, die den natürlichen Photosystemen zugrundeliegenden Prozesse zu verstehen, welche die Systeme einerseits effizient machen und andererseits vor Umwelteinflüssen schützen. Neben einer umfassenden experimentellen Erforschung bedarf es dazu einer detaillierten theoretischen Beschreibung auf molekularer, quantenmechanischer Ebene.

Zu diesem Zweck verwende ich im Rahmen meiner Arbeit die zeitabhängige Dichtefunktionaltheorie, da sie Vorhersagekraft und Verlässlichkeit mit Recheneffizienz vereint. Ihre Echtzeit-Implementierung auf einem Realraum-Gitter ist für molekulare Systeme mit hunderten oder sogar einigen tausend Elektronen geeignet. Um das volle Potential der Dichtefunktionaltheorie nutzen zu können, sind hochparallele Computer sowie Programme, die deren volle Rechenleistung ausschöpfen, erforderlich. Daher behandelt ein wesentlicher Teil meiner Arbeit die Entwicklung des Programms BTDFT, welches die zugrundeliegenden Gleichungen gitterbasiert und effizient im Zeitraum löst. Zu den wesentlichen Herausforderungen zählt hierbei die skalierbare Parallelisierung und die Entwicklung von Datenstrukturen, welche einen effizienten Speicherzugriff ermöglichen, das Programm aber gleichzeitig einfach und flexibel halten.

Für eine verlässliche Simulation von Energietransferprozessen braucht es eine genaue Beschreibung der relevanten, elektronischen Anregungen und ihres Charakters. Das bedeutet insbesondere die Vorhersage von Anregungsenergien, Dipolstärken und Übergangsdichten, welche eine Visualisierung der räumlichen Dichteoszillationen ermöglichen. Ein weiterer Teil meiner Arbeit behandelt die genaue Auswertung dieser Größen im Rahmen zeitabhängiger Dichtefunktionaltheorie im Echtzeit-Formalismus.

In der Dichtefunktionaltheorie werden alle nicht-trivialen Wechselwirkungseffekte durch das Austauschkorrelationsfunktional (xc-Funktional) beschrieben. Aufgrund der Größe typischer Lichtsammelsysteme können zu deren Beschreibung nur einfache Näherungen an das xc-Funktional verwendet werden. Diese sind zwar etabliert und wurden in der Vergangenheit mit großem Erfolg eingesetzt, weisen allerdings wohlbekannte Schwächen auf. Jene zeigen sich im Zusammenhang mit meiner Arbeit durch systematische Defizite bei der Beschreibung von Ladungstransferprozessen, was von einer spuriosen Selbstwechselwirkung eines Elektrons mit seiner eigenen Ladungsdichte herrührt. Unter Verwendung eines oder zweier gekoppelter, natürlicher Chromophore diskutiere ich im letzten Teil meiner Arbeit die Aussagekraft einer bestimmten Näherung für die Simulation von Energietransferprozessen. Dabei wird die Umgebung der Chromophore im Proteingerüst teilweise explizit oder implizit durch ein elektrostatisches Potential berücksichtigt. Abschließend gebe ich einen Ausblick auf die allgemeine Echtzeit-Simulation von Energietransferprozessen unter Verwendung der zeitabhängigen Dichtefunktionaltheorie.

Contents

1. Introduction

In view of the increasing energy demands of mankind accompanied by the pollution of our environment due to conventional power plants, the need for renewable energy sources moves more and more into the awareness of our society. This is mirrored by many public and political processes of the last years in many countries all over the world. The sun is the most important energy source for most organisms on earth. Since the incident solar energy exceeds the world's energy consumption by a factor of several thousand, its direct use appears as one of the most promising candidates for resolving the above mentioned problems. [Tur+00; Tur+12; Rog+12]

While established, inorganic solar cells are reliable and efficient they are also expensive in production. This moves the focus towards solar cells made of organic materials that are cheap to produce and highly flexible in their application, not least due to the huge variety of possibilities for engineering in organic and macromolecular chemistry. Still, at the moment, they cannot compete with their inorganic counterparts in terms of lifetime and efficiency. [GNS07; TF08; Hag+10]

Nature has perfected its light-harvesting mechanisms in plants, algae, and special kinds of bacteria. Photosynthetic purple bacteria, as an outstanding example, convert the incident light energy with an efficiency of nearly 100 %. Furthermore, they developed effective photoprotection techniques that prevent the related macromolecular complexes from being damaged by the incoming radiation. It is not surprising that the investigation of the underlying biological, chemical, and physical processes preoccupies natural science since many years, not least in view of applications in artificial light harvesting. At the center of this discussion are questions concerning the occurrence of quantum-mechanical coherence in the excitation-energy transfer (EET) or the separation of charges within the reaction center (RC). [Küh95; Hu+02; CGK06; Eng+07; CF09; Fle+12; SŞS12]

To understand the related, complex processes that ultimately lead to the observed efficiency and stability of natural light harvesting, a theoretical description by means of quantum mechanics is mandatory. The size of light-harvesting systems with thousands of electrons, the biological environment, and the underlying processes taking place on various length and time scales makes this task challenging for state-of-the-art methods. Moreover, theory cannot simulate nature in its full complexity but relies on model assumptions and approximations, which typically have a narrow scope. It is therefore necessary to combine different methods to answer distinct aspects of the fundamental questions, which may be merged into an image of reality. [Hu+02; CF09; CM16]

At the lower end of length and time scales, this begins with the molecules that are the key building blocks of natural light-harvesting systems. Investigating the dynamics of electronic excitations therein requires a powerful machinery of many-body quantum theory and its implementation through efficient program codes that are optimized for modern, highly parallel computers. For this task time-dependent density functional theory (TDDFT) is a natural candidate since it combines a first principles approach with reliability, computational efficiency, and flexibility. In particular, its efficient real-time formulation as well as the increasing computational power bring natural light-harvesting systems into the reach of modern computational physics. [YB96; CRS97; SS96; YB99; CMR04; Kro+06; And+15; Jor+15]

During my work, I developed a program called BTDFT that performs calculations in

the real-space and real-time framework of TDDFT. In view of large molecular systems with up to a few thousand electrons, I laid special emphasis on an scalable parallelization and an efficient memory-access. Since the proper description of excitations and excited states lies at the heart of EET simulations, I elaborated a scheme to obtain those data accurately from real-time propagations. I used this scheme to investigate the reliability and predictive power of real-time TDDFT for the description of EET between the chromophores in the LH2 antenna complex of purple bacteria. In the focus of my study were the occurrence of spurious excitations, the role of the electrostatic environment of the chromophores, and the proper description of real-time energy transfer.

This thesis is organized as follows: In section 2 I introduce the main concepts of TDDFT. These include the fundamental theorems of Hohenberg and Kohn as well as Runge and Gross and the Kohn-Sham formulation of TDDFT. Subsequently, I present the BTDFT program in section 3. After a short excursus about modern computer architectures, I explain the basic grid setup, the parallelization, and the most important principles and algorithms. I finally discuss one of the fundamental approaches for the performance engineering on BTDFT and show performance tests for practically relevant systems. More details on the implementation, compilation, execution, etc. of BTDFT can be found in appendix A. General remarks on simple performance engineering approaches are summarized in appendix C. The subsequent discussion about the real-time description of EET requires a detailed knowledge of excited states in the relevant energy range. Therefore, I elaborate a scheme to obtain accurate excitation energies, oscillator strengths, and transition densities from real-time TDDFT in section 4. Finally, section 5, which reports the current state of my research, is dedicated to the dynamics of excitation energy between chromophores that appear in the LH2 antenna complex of *Rhodoblastus* (*Rbl.*) *acidophilus*. In this course, I introduce the relevant structural features of the chromophore-protein complex and discuss the reliability of EET simulations with real-time TDDFT in the so-called B850 ring by means of one and two aggregated bacteriochlorophylls (BChl). The latter includes the effect of the BChl's environment, which is treated by an electrostatic potential or partly explicitly within the TDDFT calculation. I further investigate the computation of coupling strengths from real-time simulations and discuss the direct use of density fluctuations as a measure for an energy density or an energy per chromophore in view of future EET simulations.

2. Density functional theory and time-dependent density functional theory

2.1. Quantum mechanics and the exponential wall

Quantum mechanics is the fundamental theory for the description of microscopic systems. While there exist many philosophical interpretations of quantum mechanics concerning, e.g., determinism or the role of measurement processes, its mathematical evaluation and application is, in principle, clear. All information about a stationary system is contained in an abstract wave function $|\psi\rangle$ that obeys the stationary Schrödinger equation [Sch26]

$$\hat{H}|\psi\rangle = E|\psi\rangle \tag{2.1}$$

with the energy

$$E = \langle\psi|\hat{H}|\psi\rangle \tag{2.2}$$

and the Hamiltonian $\hat{H}$. For a system in an external potential $v(\boldsymbol{r})$ with a two-particle interaction $w(\boldsymbol{r}, \boldsymbol{r'})$ the Hamiltonian in its spatial representation reads

$$\hat{H} = \underbrace{\sum_{j=1}^{N} -\frac{\hbar^2}{2m}\Delta_{\boldsymbol{r_j}}}_{\hat{T}} + \underbrace{\sum_{j=1}^{N} v(\boldsymbol{r_j})}_{\hat{V}} + \underbrace{\frac{1}{2}\sum_{\substack{j,k=1 \\ j\neq k}}^{N} w(\boldsymbol{r_j}, \boldsymbol{r_k})}_{\hat{W}} \,. \tag{2.3}$$

It can be split into operators for the kinetic energy $\hat{T}$, the potential energy $\hat{V}$, and the interaction energy $\hat{W}$. N is the number of particles and $\Delta_{\boldsymbol{r_j}}$ is the Laplacian that acts on the coordinate $\boldsymbol{r_j}$ of the j^{th} particle. m is the particle's mass and $\hbar$ is Plank's constant. Expectation values of observables, which are described by an operator $\hat{O}$, are calculated via

$$\langle\hat{O}\rangle_\psi = \langle\psi|\hat{O}|\psi\rangle = \int \dots \int \psi^*(\{x_j\})\hat{O}\psi(\{x_j\})dx_1 \dots dx_N \,, \tag{2.4}$$

where $\{x_j\}$ is a shorthand notation for $(\boldsymbol{r_1}, s_1), \dots, (\boldsymbol{r_N}, s_N)$ and s_j is the spin index. The integral over dx_j includes a spatial integration over $d^3\boldsymbol{r_j}$ and a summation over the spin values s_j.

From equation (2.1) and (2.3) it is clear that the wave function in its spatial representation depends on the coordinates of all N particles. The analytical solution of the Schrödinger equation is impossible in almost all cases and the effort for its numerical solution rises exponentially with the number of particles. This is commonly known as 'exponential wall' [Koh99, p. 1257].

The complexity of the wave function makes approximations necessary for practical applications. On the one hand, many of the earlier approaches such as the one of Heitler and London [HL27] or the one of Hartree and Fock [Har28; Foc30] approximated the wave function itself. On the other hand, Thomas-Fermi theory [Tho27; Fer27], even if not very successful in many respects [Cap06; Bur07], can be considered as the first, approximate description of quantum mechanics that only uses the particle density as the central object.

2.2. Kohn-Sham density functional theory

2.2.1. The Hohenberg-Kohn theorem and variational principle

Modern density functional theory (DFT) is a reformulation of quantum mechanics that relies on the ground-state particle density $n_0(\boldsymbol{r})$ alone. The fundamental statement of DFT for non-degenerate systems is that there exists a one-to-one correspondence between the ground-state density and the external potential, up to an additive constant. As a consequence, the wave function and therefore all ground-state observables are unique functionals of the ground-state density, i.e., $|\psi\rangle = |\psi[n_0]\rangle$ and $O[n_0] = \langle\hat{O}\rangle_{\psi[n_0]}$. This was proven by Hohenberg and Kohn (HK) in 1964 [HK64] and is therefore termed HK theorem. Deeper insights into the contents of the current section are subject of other reviews and books, e.g., [DG90; Koh99; Fio+03; PK03; Cap06; Bur07].

The proof of the HK theorem employs that the Hamiltonian in equation (2.3) is, for a given two-particle interaction $w(\boldsymbol{r}, \boldsymbol{r'})$, uniquely determined by the potential $v(\boldsymbol{r})$ and the number of particles N. The particle number is obtained from the ground-state density by spatial integration and thus a simple functional of the density. Hohenberg and Kohn showed in their proof that the same is true for the potential $v = v[n_0]$ and thus for the whole Hamiltonian and its eigenvalues and eigenfunctions.

The second statement, termed HK variational principle, says that the energy $E_v[n] = \langle\hat{T} + \hat{V} + \hat{W}\rangle_{\psi[n]}$ is as well a functional of the density and minimized by the GS density n_0. $E_0 = E_v[n_0] = \min_{\int n(\boldsymbol{r})d^3r = N} E_v[n]$ is the corresponding ground-state energy where the minimization is constrained to the densities that result from a valid potential $v(\boldsymbol{r})$ and integrate to the particle number N.

The HK theorem, on the one hand, guarantees the uniqueness of a system that corresponds to a given ground-state density. The HK variational principle, on the other hand, is a first step towards its application since it sets a route to determine the GS density from the minimization of the energy functional. The latter is not known in general but can be divided into two parts

$$E_v[n] = \underbrace{\langle\hat{V}\rangle_{\psi[n]}}_{=V[n]} + \underbrace{\langle\hat{T} + \hat{W}\rangle_{\psi[n]}}_{=F_{\mathrm{HK}}[n]} . \tag{2.5}$$

The first functional reads

$$V[n] = \int n(\boldsymbol{r}) v(\boldsymbol{r}) d^3r \tag{2.6}$$

and is easily evaluated for a given density and potential. The HK functional $F_{\mathrm{HK}}[n]$ is unknown but only depends on the kind of two-particle interaction. Hence, $F_{\mathrm{HK}}[n]$ is the same for all systems with the same two-particle interaction and is universal in this respect.

The restriction to non-degenerate systems can be lifted. The only difference is that the functional dependence of an observable may not be unique if two degenerate states provide the same density. All other statements remain, especially that the ground-state energy remains a functional of the density as well as the HK variational principle. [DG90]

2.2.2. The Kohn-Sham equations

The statement of the HK theorem that all information about a time-independent system is given by its ground-state density is remarkable. Further does the HK variational principle set a route for obtaining the ground state from an unknown but approximable energy functional by minimizing it with respect to valid N-particle densities. The dawn of DFT, however, came with its Kohn-Sham (KS) formulation [KS65], which made DFT applicable to a broad scope of applications while delivering results with high accuracy and computational efficiency [DG90]. The contents in this section again follow [DG90].

Kohn and Sham formulated DFT using an auxiliary system of N non-interacting particles that move in an effective, local potential v_{KS} and reproduce the density of the interacting system. The KS particles follow the equation

$$\hat{H}_{\mathrm{KS}}\varphi_j(\boldsymbol{r}) = \varepsilon_j\varphi_j(\boldsymbol{r})\,, \quad \varepsilon_1 \leq \varepsilon_2 \leq \ldots \tag{2.7}$$

with the KS Hamiltonian

$$\hat{H}_{\mathrm{KS}} = -\frac{\hbar^2}{2m}\Delta + v_{\mathrm{KS}}(\boldsymbol{r})\,, \tag{2.8}$$

the KS orbitals φ_j, and KS eigenvalues ε_j for $j = 1, \ldots, N$. Since the HK theorem guarantees that v_{KS} is unique[1], the KS orbitals are functionals of the density $\varphi_j = \varphi_j[n]$. The density of the KS system, and therefore the density of the real interacting one, is given by

$$n(\boldsymbol{r}) = n_{\mathrm{KS}}(\boldsymbol{r}) = \sum_{j=1}^{N} |\varphi_j(\boldsymbol{r})|^2\,. \tag{2.9}$$

The resulting equations look like those of an effective single-particle theory. Still, the reformulation of quantum mechanics in the KS scheme is in principle exact if the exact KS potential v_{KS} is chosen. In order to derive an expression for v_{KS}, one applies the HK variational principle to the energy, which is split into

$$E_v[n] = T_{\mathrm{KS}}[n] + \int v(\boldsymbol{r})n(\boldsymbol{r})d^3r + \frac{1}{2}\int n(\boldsymbol{r})w(\boldsymbol{r},\boldsymbol{r'})n(\boldsymbol{r'})d^3r d^3r' + E_{\mathrm{xc}}[n]\,. \tag{2.10}$$

$T_{\mathrm{KS}}[n]$ is the kinetic energy of the non-interacting KS system

$$T_{\mathrm{KS}}[n] = -\int \frac{\hbar^2}{2m}\sum_{j=1}^{N}\varphi_j^*(\boldsymbol{r})\Delta\varphi_j(\boldsymbol{r})d^3r\,, \tag{2.11}$$

which is a valid density functional since the KS orbitals are valid density functionals. For interacting electrons, w is the Coulomb interaction $w(\boldsymbol{r},\boldsymbol{r'}) = \frac{e^2}{4\pi\varepsilon_0}\frac{1}{|\boldsymbol{r}-\boldsymbol{r'}|}$ and the third term on the right hand side of equation (2.10) is the Hartree energy [Har28; Cap06; DG90]

$$E_{\mathrm{H}}[n] = \frac{e^2}{8\pi\varepsilon_0}\int \frac{n(\boldsymbol{r})n(\boldsymbol{r'})}{|\boldsymbol{r}-\boldsymbol{r'}|}d^3r d^3r'\,. \tag{2.12}$$

[1] Implications of this statement are discussed in [DG90] under the heading "v-representability".

The so-called exchange-correlation (xc) energy

$$E_{\mathrm{xc}}[n] = F_{HK}[n] - T_{\mathrm{KS}}[n] - \frac{1}{2}\int n(\boldsymbol{r})w(\boldsymbol{r},\boldsymbol{r'})n(\boldsymbol{r'})d^3r d^3r' \tag{2.13}$$

contains all remaining many-particle effects.

Employing the HK variational principle to the energy (2.10) results in an expression for the KS potential

$$v_{\mathrm{KS}}(\boldsymbol{r}) = v(\boldsymbol{r}) + v_{\mathrm{H}}(\boldsymbol{r}) + v_{\mathrm{xc}}(\boldsymbol{r}) \tag{2.14}$$

with the Hartree potential

$$v_{\mathrm{H}}(\boldsymbol{r}) = \frac{e^2}{4\pi\varepsilon_0}\int \frac{n(\boldsymbol{r'})}{|\boldsymbol{r}-\boldsymbol{r'}|}d^3r' \tag{2.15}$$

and the xc potential

$$v_{\mathrm{xc}}(\boldsymbol{r}) = \frac{\delta E_{\mathrm{xc}}[n]}{\delta n(\boldsymbol{r})}\,. \tag{2.16}$$

Equations (2.7), (2.9), and (2.14) define the KS equations that are usually solved self-consistently. The xc energy typically has to be approximated in practice. The xc approximations used in this thesis are outlined in section 2.5.

Finally, the KS eigenvalues and KS orbitals have been introduced as purely mathematical objects and one has to be careful in interpreting them in terms of physical quantities. However, at least some of them have an important meaning (see e.g. [DG90; Cap06]). The eigenvalue of the highest occupied molecular orbital (HOMO) of an N-electron system equals minus the first ionization energy $IP(N) = E_0(N-1) - E_0(N)$. The HOMO eigenvalue of the system with $N+1$ electrons equals minus the electron affinity $EA(N) = E_0(N) - E_0(N+1)$, i.e., [Per+82; LPS84; AB85; Cap06]

$$\varepsilon_{\mathrm{HOMO}}(N) = -\mathrm{IP}(N) \quad \text{and} \quad \varepsilon_{\mathrm{HOMO}}(N+1) = -\mathrm{EA}(N)\,. \tag{2.17}$$

$E_0(M)$ from above is the ground-state energy of the respective M-electron system. In addition, the density is asymptotically dominated by the orbital density of the HOMO[2] [Kre+98], which is therefore important for the electronic structure of a molecule as it is seen from outside.

The above equalities are only exact if the exact xc energy functional is applied. Since usually approximations have to be used, the approximate $\varepsilon_{\mathrm{HOMO}}$ may often be an unsatisfying approximation to the real IP or EA [Cap06].

2.3. Time-dependent Kohn-Sham density functional theory

2.3.1. The Runge-Gross theorem

Time-dependent density functional theory (TDDFT) [ZS80; RG84; CAS95; PGG96; GDP96; Lee98; Lee01; MG04; Mar+12] is an extension of the ideas of DFT to time-dependent systems. In analogy to its time-independent counterpart, TDDFT states that the time-dependent particle density $n(\boldsymbol{r},t)$ determines the time-dependent potential $v(\boldsymbol{r},t)$ up to a purely time-dependent function. The related proof was formulated

[2] But on nodal surfaces of the HOMO [KP03].

by Runge and Gross (RG) in 1984 [RG84]. The contents of this section follow [MG04; GDP96].

In TDDFT the role of the energy (2.2) is replaced by the action

$$A[\psi] = \int_{t_0}^{t_1} \langle \psi(t) | i\hbar \frac{\partial}{\partial t} - \hat{H}(t) | \psi(t) \rangle \tag{2.18}$$

with the Hamiltonian of equation (2.3) but a time-dependent external potential $v = v(\boldsymbol{r}, t)$. The wave function that makes the action stationary is the solution of the time-dependent Schrödinger equation. At its stationary point, the action is always zero. This makes the action a much less useful quantity compared to the energy in the time-independent case since the action at its stationary point contains no physical information. Moreover, the replacement of the minimum principle by a stationary principle makes the proof of the RG theorem more elaborate than that of the HK theorem. In addition, the stationary point of the action, and therefore the time-dependent density and all related quantities, depends on the initial state of the system since the time-dependent Schrödinger equation is an initial value problem. This means that the one-to-one correspondence between the time-dependent density and the time-dependent potential in the RG theorem holds only for a fixed initial state.

2.3.2. The time-dependent Kohn-Sham equations

The time-dependent Kohn-Sham (TDKS) scheme seems to be a straightforward extension of the ground-state ideas. The KS orbitals follow the effective single-particle equation [MG04; GDP96]

$$i\hbar \frac{\partial}{\partial t} \varphi_j(\boldsymbol{r}, t) = \underbrace{\left[-\frac{\hbar^2}{2m} \Delta + v_{\mathrm{KS}}(\boldsymbol{r}, t) \right]}_{= \hat{H}_{\mathrm{KS}}(t)} \varphi_j(\boldsymbol{r}, t) \tag{2.19}$$

with the KS potential

$$v_{\mathrm{KS}}(\boldsymbol{r}, t) = v(\boldsymbol{r}, t) + v_{\mathrm{H}}(\boldsymbol{r}, t) + v_{\mathrm{xc}}(\boldsymbol{r}, t)\,. \tag{2.20}$$

The time-dependent density is

$$n(\boldsymbol{r}, t) = \sum_{j=1}^{N} |\varphi_j(\boldsymbol{r}, t)|^2 \tag{2.21}$$

and the time-dependent Hartree potential is the one of equation (2.15) evaluated with $n(\boldsymbol{r}, t)$.

However, the time-dependent xc potential is more involved. Similar to the time-independent case, v_{xc} can be defined as the functional derivative of an xc action functional with respect to the time-dependent density. However, this leads to an inconsistency concerning causality [GDP96, §5.1] that was resolved by van Leeuwen [Lee98] by defining a new action functional $\tilde{A}$ within the Keldysh formalism. The time-dependent xc potential can thus be expressed as

$$v_{\mathrm{xc}}(\boldsymbol{r}, t) = \left. \frac{\delta \tilde{A}_{\mathrm{xc}}}{\delta n(\boldsymbol{r}, \tau_{\mathrm{K}})} \right|_{n(\boldsymbol{r}, t)} \tag{2.22}$$

with the Keldysh pseudo time τ_{K}.

A further, practical problem is that finding good approximations for the action is cumbersome. Instead, it is common practice to use an adiabatic approximation. To this end, the time-dependent density is inserted into an approximate ground-state xc functional. Adiabatic approximations are local in time and therefore neglect memory effects. [GDP96; Fio+03; MG04]

2.4. Spin density functional theory in spin-unpolarized systems

The electronic systems addressed in this thesis have an additional spin degree of freedom that did not appear so far. Still, all presented theorems can easily be extended to spin-dependent systems [BH72; RC73; DG90; PK03; Fio+03]. In this respect, the total density is decomposed into the spin densities of the two spin channels up ($\uparrow$) and down ($\downarrow$)

$$n = \sum_{\sigma=\uparrow,\downarrow} n_\sigma = \sum_{\sigma=\uparrow,\downarrow} \sum_{j=1}^{N_\sigma} |\varphi_{j\sigma}|^2 \,, \tag{2.23}$$

where N_σ is the number of occupied spin orbitals $\varphi_{j\sigma}$ in the spin channel σ. The density and the orbitals can be time-dependent or time-independent in this equation.

All (TD)KS equations remain essentially the same. Yet, the xc energy (or action) now depends on both spin densities rather than the total density alone, i.e., $E_{\mathrm{xc}} = E_{\mathrm{xc}}[n_\uparrow, n_\downarrow]$. In ground-state DFT the spin-dependent xc potential is therefore defined as

$$v_{\mathrm{xc},\sigma}(\boldsymbol{r}) = \frac{\delta E_{\mathrm{xc}}[n_\uparrow, n_\downarrow]}{n_\sigma(\boldsymbol{r})} \,. \tag{2.24}$$

The same applies to the time-dependent case.

The KS orbitals, the KS eigenvalues, the KS potential, and therefore the KS Hamiltonian get an additional spin index. This means that the (TD)KS equations must in principle be solved for both spin channels, which are coupled by the Hartree and xc potentials. However, if the system has no spin polarization, i.e., both spin channels are equally occupied and have equal spin densities, only one spin channel needs to be calculated. This is the case in all of the calculations presented in this work.

2.5. Approximations to the exchange-correlation functional

The sections about the KS formulation of ground-state DFT and TDDFT introduced the concept of exchange and correlation, which is also known from other many-particle quantum theories such as Hartree-Fock [Har28; Foc30] and successors. A key issue in DFT is to choose a meaningful approximation for the xc energy (or action) functional, which leads to the xc potential appearing in the (TD)KS equations. At the current stage of the theory, many xc functionals have been developed that improve the predictive power of (TD)DFT but also increase the computational effort.

The development of new xc functionals has been described by John Perdew by means of Jacob's ladder [PK03; KK08]: On the first rung stands the local density approximation (LDA) [KS65], which depends on the density in a spatially local way. Each further rung adds complexity to the functionals.

Including the gradient of the density leads to the semi-local generalized gradient approximations (GGA), of which the one of Perdew, Burke, and Ernzerhof (PBE) [PBE96; PBE97] is very prominent. So-called orbital functionals [KK08] are expressed through the KS orbitals $E_{\mathrm{xc}} = E_{\mathrm{xc}}[\{\varphi_j[n]\}]$ and thus depend on the density in a non-local way. The exact exchange (EXX) [GG95; GKG97], which is the Hartree-Fock exchange evaluated with the KS orbitals, and self-interaction corrected (SIC) [PZ81] functionals are of this type. Both counteract the so-called self-interaction error discussed in a subsequent section. [Cap06; KK08]

A further step in the development of new functionals is to mix the EXX functional with semi-local exchange and correlation, which results in so-called global hybrids such as B3LYP [Bec93; Ste+94]. Finally, range-separated hybrid (RSH) [BN05; SKB09b; SKB09a; BLS10; Kar+11; Kör+11; Kro+12; KKK13] functionals such as ωPBE [VS06] split the interaction into a long-range and a short-range part, which are treated with different approximations for exchange and correlation.

Evaluating the xc potential from orbital functionals through equation (2.16) requires the solution of the optimized effective potential (OEP) equation [SH53; TS76; KLI92a; KLI92b; EV93; LKI93; GG95; GKG97]. This is cumbersome in ground-state DFT and almost impossible in TDDFT [KK08; WU08; Sch13]. Therefore, approximations by, e.g., Krieger, Li, and Iafrate (KLI) [KLI90] or localized Hartree-Fock (LHF) [DG01; GGB02] are widely used. An alternative scheme that circumvents the OEP equation uses orbital-specific xc potentials and requires a generalized KS (GKS) framework [Sei+96].

The list of ground-state xc functionals given above is incomplete but shows the variety of possibilities, each with strengths and weaknesses [Fio+03; PK03; KK08]. In TDDFT things usually get even more involved since the exact xc action functional is known to be non-local in both, space and time [Fio+03; PK03; KK08]. However, adiabatic approximations, which are local in time, have demonstrated their usefulness in practice [Cal+00; Fio+03]. In the course of this thesis, the adiabatic time-dependent LDA (TDLDA) and the optimally tuned ωPBE are of importance. In the following, I briefly introduce those two, especially in view of the self-interaction error and the description of charge-transfer excitations.

2.5.1. The local density approximation

The LDA [KS65] is one of the most important and basic xc functionals. The LDA exchange energy is given by

$$E_x^{\mathrm{LDA}}[n] = \int e_x^{\mathrm{LDA}}[n]\big|_{n(\boldsymbol{r})}\, d^3r\,, \tag{2.25}$$

where

$$e_x^{\mathrm{LDA}} = -\frac{3e^2}{16\pi\varepsilon_0}\left(\frac{3}{\pi}\right)^{\frac{1}{3}} n^{\frac{4}{3}} \tag{2.26}$$

is the exchange energy density of the homogeneous electron gas with constant density n. For the correlation energy density of the homogeneous electron gas, one uses a parametrized expression that satisfies exact constraints such as the one of Perdew and Wang [PW92]. The parameters are then fitted to data from accurate quantum Monte Carlo simulations [CA80]. In the time-dependent case the TDLDA functional uses the

same expression but is evaluated with the time-dependent density in the sense of an adiabatic approximation. [GDP96; Fio+03; MG04; Cap06]

Due to the simplicity of the (TD)LDA, one is tempted to think that it only performs well for systems with a density that varies slowly in space and time. However, (TD)LDA performs reasonably well in many situations and is computationally efficient. [Cap06]

It still has well known deficiencies. Some of those lead to quantitative errors as for the prediction of chemical bond lengths, which can often be improved by using GGA functionals [Cap06; KK08]. Others, such as the wrong description of charge transfer and charge-transfer excitations [LLS03; Toz03; DH04; Moo+15], are of fundamental nature and wrongly described by all typical (semi-)local functionals [KK08]. The latter is related to the self-interaction error, which is discussed in the following.

2.5.2. Self interaction and charge transfer

The self-interaction error [KK08] is the reason for many deficiencies of (semi-)local xc functionals. It is based on the splitting of the interaction energy into a Hartree part (2.12) and an xc part. In the one-electron limit there is no electron-electron interaction and the xc part of the energy in equation (2.10) must cancel the Hartree contribution. However, typical (semi-)local xc functionals do not fulfill the requirement of being free from one-particle self interaction as defined in [PZ81]. This leads to electrons interacting with their own charge density. [KK08]

Typical symptoms that are related to the presence of self-interaction are a wrong asymptotic behaviour of the xc potential [PZ81; Kre+98; Fio+03; KK08], a missing derivative discontinuity [Per+82; CMY08; KK08] of the xc energy, and a missing field-counteracting effect of the xc potential [Gis+99; KK08; HK12]. In practice, this leads to a wrong description of, among others, charge-transfer processes [KK08] and charge-transfer states [LLS03; Toz03; DH04; Moo+15]. As a simplified example, consider one electron sitting in one of two potential wells that are separated by a finite barrier, e.g., two separated molecules. Since an electron is repelled by its own charge density due to the spurious self interaction, the movement from one site to the other, i.e., the related charge transfer, is much easier than it should be without the self-interaction error. In molecules, the excitation energies of excitations that show a charge-transfer character are usually underestimated substantially [DH04; SKB09b].

An expression that is known to cancel the one-particle self-interaction error exactly is the Fock exchange energy or EXX [KK08]

$$E_x^{\text{exact}} = -\frac{e^2}{8\pi\varepsilon_0} \sum_{\sigma=\uparrow,\downarrow} \sum_{j,k=1}^{N_\sigma} \int\int \frac{\varphi_{j,\sigma}^*(\boldsymbol{r})\varphi_{k,\sigma}^*(\boldsymbol{r'})\varphi_{j,\sigma}(\boldsymbol{r'})\varphi_{k,\sigma}(\boldsymbol{r})}{|\boldsymbol{r}-\boldsymbol{r'}|} d^3r d^3r' \,, \tag{2.27}$$

which is evaluated with the KS spin orbitals. The EXX functional corrects many of the above mentioned issues such as the asymptotic behaviour of the xc potential and the missing derivative discontinuity of the xc energy but completely lacks correlation. However, it is not straightforward to add correlation in a compatible way [KK08]. The approach of a global hybrid is to mix semi-local exchange and EXX with a global parameter while treating correlation in a completely semi-local way. This leads to a

useful form of a hybrid functional [KK08]

$$E_{\mathrm{xc}}^{\mathrm{hybrid}} = bE_x^{\mathrm{exact}} + (1-b)E_x^{\mathrm{approx}} + E_c^{\mathrm{approx}} \tag{2.28}$$

with a global parameter b that is usually fitted to a set of test systems. E_x^{exact} of the ansatz above denotes the EXX functional whereas E_x^{approx} and E_c^{approx} are the exchange and correlation energies of a semi-local approximation. Modern hybrid functionals such as B3LYP [Bec93; Ste+94] usually have a more complex form.

2.5.3. Range-separated hybrid functionals

The idea of RSH functionals [BN05; SKB09b; SKB09a; BLS10; Kar+11; Kör+11; Kro+12; KKK13] is to split the Coulomb interaction into a long-range and a short-range contribution [SF95; Lei+97]

$$\frac{1}{r} = \underbrace{\frac{f_\omega(r)}{r}}_{\text{Long-range}} + \underbrace{\frac{1-f_\omega(r)}{r}}_{\text{Short-range}} . \tag{2.29}$$

$f_\omega(r)$ weights the long-range and short-range parts of the interaction and typically depends on a range-separation parameter ω. One common choice for $f_\omega(r)$ is the error function, i.e., $f_\omega(r) = \mathrm{erf}(\omega r)$ [Kör+11].

The parameter ω determines the transition from short-range interaction to long-range interaction and introduces a characteristic length scale $\frac{1}{\omega}$. The splitting of the Coulomb interaction in equation (2.29) provides the possibility to combine the advantages of semi-local (or hybrid) exchange in the short-range part with the long-range behaviour of EXX [Kör+11].

Since the RSH approach can be applied with different semi-local or hybrid exchange functionals for the short-range part, one can think of the RSH functional in terms of a long-range correction [VS06; Kör+11]. This typically resolves the wrong asymptotic behaviour from the underlying semi-local exchange functional and counteracts the self-interaction error. In particular, RSH functionals are typically well suited for the description of charge transfer [SKB09b] and charge-transfer excitations [SKB09a; Kar+11]. Thus, they resolve one of the major deficiencies of (TD)DFT, at least partly [LB07]. Nevertheless, RSH functionals again have fundamental issues like the violation of size consistency [Kar+11], which is not part of this thesis.

There exist two major approaches to determine the range-separation parameter ω. The first one is to use a value that is optimized for the description of a set of test systems [CH08]. The second one is to perform a tuning procedure for every single system, which is known as optimal tuning [LB07; SKB09b; SKB09a; Kör+11; Kar+11; KKK13]. To this end, ω is usually chosen to fulfill an exact property such as the IP theorem from equation (2.17) [SKB09b; SKB09a]. This requires a self-consistent procedure and the solution of possibly many DFT calculations with different values of ω.

While the first option is computationally less demanding, it is known that the optimal value of ω can strongly depend on the system under consideration. The optimal tuning approach, which is employed in the calculations presented in this thesis, usually delivers results with a much better quality. [SKB09b; SKB09a; Kör+11]

In the course of this thesis, the PBE functional was chosen for the semi-local part. This results in the RSH functional ωPBE [VS06]. Finally, one should keep in mind that the RSH functionals are usually treated within the framework of GKS [Sei+96]. This is also the case for the ωPBE calculations presented in section 5 of this work.

3. The BTDFT program set

3.1. Introductory notes

All real-space and real-time calculations presented in this thesis have been done with a program called BTDFT (Bayreuth density functional theory)[3]. It is solely written in Fortran in the 2003 standard [Ada+09] and contains some object-oriented elements to structure the code. Programs with a similar scope are PARSEC [Kro+06] and Octopus [Mar03a; Cas+06; And+15].

I developed BTDFT as a major part of my PhD, so I dedicated an own section to it. The aim of BTDFT is to solve the (TD)KS equations of sections 2.3.2 and 2.2.2 on a real-space grid in real time with equidistant grid and time spacings. The first goal of this project was to create an easily extendable program that establishes an efficient framework for further development of (TD)DFT. The second goal was to provide a fast program for studying large-scale quantum systems within (TD)DFT in real space and real time. In this spirit I laid special emphasis on an efficient parallelization and memory-access as well as a modular code structure that allows for the simple implementation of extensions.

The most striking differences to PARSEC are an additional orbital parallelization and the way of applying semi-local operators such as the Laplacian or Hamiltonian, on which the program spends most of its computation time. Apart from that, I used different algorithms for, e.g., the real-time propagation. While I introduce most of the ideas by means of simple models and example calculations, I present performance tests on two relevant molecular systems in section 3.5. One outcome of the tests is that BTDFT can perform a propagation in less than 5% of the time required by PARSEC. Using an improved propagation algorithm and the additional orbital parallelization results in additional performance boosts. Hence, my hope and expectation is that BTDFT will contribute to the study of natural light harvesting and other fields of interest in the future.

During the development, several group members have been using BTDFT and already have implemented their own work into the code. Among the current capabilities of BTDFT, which are not part of this work, are the simulation of AFM images [Sch16], charge transfer [Sch16], and angle-resolved photoemission spectroscopy (ARPES) [Dau16; Dau+16]. To this end, a variety of time-dependent external potentials such as classical electric fields with linear or circular polarization, a Förster-type Hartree potential [HKK10; Ber16], absorbing and anti-absorbing boundary conditions [Sch16; SK16], or an external environment potential (see section 5.2.1) have been implemented. In the current version of BTDFT (Nov. 2016) the (TD)LDA [KS65; RG84; PW92] xc functional as well as (TD)EXX [GG95; GKG97; KK08] and KS (TD)LDA-SIC [PZ81; KK08; KKM08][4] in the KLI [KLI90] and LHF [DG01; GGB02] approximations are supported.

BTDFT comprises five programs: BTDFT_guess, BTDFT_gs, BTDFT_td, ace2human, and parsec2ace. BTDFT_guess, BTDFT_gs, and BTDFT_td make an initial guess for the density of a given atom setup, calculate the ground state of the system by solving the KS equations, and finally propagate the system in time. These three

[3] In former thesis BTDFT is referred to as yACES, which was its unofficial name.
[4] Currently without unitary orbital transformations [KKM08; HKK12; HK12; Hof+12].

programs communicate with each other through files in a special binary format I called ACE. The splitting into individual programs has practical reasons, which is explained in appendix A in more detail.

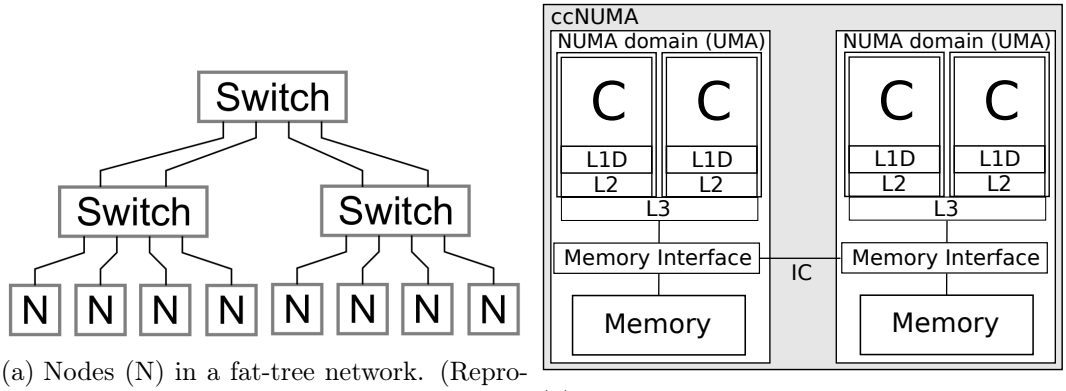
The program ace2human reads ACE files and outputs their contents, such as the density, into human readable form. Gaussian cube files [Fri+], which are widely used to store data on real-space grids, are also supported. parsec2ace, on the other hand, converts a special PARSEC output file into an ACE file that can be used as input for a propagation with BTDFT_td. To get the appropriate PARSEC output, I implemented an interface Fortran module that can be added to the PARSEC code.

The programs BTDFT_gs and BTDFT_td are parallelized with the Message Passing Interface (MPI) [MPI12]. To keep the communication between MPI processes efficient, I used non-blocking communication, derived MPI types for non-contiguous send and receive buffers, virtual process topologies, and parallel input and output (IO). Some of the underlying ideas are introduced or mentioned in the following sections.

Besides the source code, the project consists of various files that are used for compiling and running the code as well as a Doxygen [Hee16] documentation. The file tree is explained in appendix A.1 and is under git version control (see appendix A.6). Additional information can be found in appendix A and in the Doxygen documentation introduced in appendix A.5.

Since the goal of my work was to develop an efficient program for (TD)DFT calculations, I explain the most important considerations in this respect. In particular, section 3.2 is about modern computer architectures, especially in view of the computers and the MPI parallelization used. In section 3.3 I introduce the real-space grid of BTDFT and the grid parallelization as well as the additional orbital parallelization. Moreover, I discuss the most important algorithms and approaches used for the solution of the (TD)KS equations and the Hartree potential. Section 3.4 is dedicated to some of the performance engineering. Specifically, I introduce a simple memory-access model and show how BTDFT applies the Hamiltonian operator. The latter has a strong impact on the overall performance of the code. Further, more general performance considerations are discussed in appendix C. Finally, I show two test cases in section 3.5, particularly regarding the total performance of BTDFT and the additional orbital parallelization.

For more information I recommend the following literature, which was the basis of my work: General algorithms such as the fast Fourier transform (FFT), the Crank-Nicolson propagator, or finite differences schemes are found in the Numerical Recipes [Pre+92]. Iterative solver algorithms for linear systems of equations are well explained in [Mei11]. MPI is defined in the MPI standard [MPI12] and a tutorial is given in [Pac96]. Modern computer architectures and an introduction to performance models and performance engineering is given in [HW10]. For performance benchmarking and related utilities I used the STREAM benchmarks [McC95], the LIKWID [THW10] program package, which is related to the authors of the latter book, and the OMB (OSU Micro Benchmarks) [Pan16] package for MPI.



(a) Nodes (N) in a fat-tree network. (Reproduced from [HW10, Fig. 4.16] in a similar fashion)

(b) Schematic of a ccNUMA node with two sockets and two cores (C) each. (Reproduced from [HW10, Fig. 4.5] in a similar fashion)

Figure 3.1: Sketch of a modern supercomputer with ccNUMA nodes in a fat-tree network.

3.2. Modern computer architectures and parallelization

3.2.1. Modern computer architectures

A natural point to start the discussion about writing efficient program code is to talk about modern computer architectures. Many of today's supercomputers, such as btrzx3 and btrzx5 at the University of Bayreuth (UBT), consist of nodes that are connected by a communication network. An example is the fat-tree network shown in figure 3.1a. In this fat-tree network, which is only one out of many possibilities [HW10, §4.5], the nodes are the leaves that are connected by switches in a tree-like manner. Each node has its own operating system and main memory that can usually not directly be accessed by a remote node. One therefore speaks of a distributed memory computer [HW10, §4.3] on this level. [HW10, §4]

A today's node usually consists of a certain number of physical processors, which are mounted on sockets on the node and connected by an coherent interconnection (IC) network [HW10, §4.2.3]. Figure 3.1b shows an example of such a node with two multi-core processors with two cores (C) each. Each core consists of a separate L1 data (L1D) and L2 cache as well as controllers, registers, and functional units that are responsible for integer and floating point operations (not shown). In this example both cores within the same processor have an additional, shared L3 cache that is connected to the main memory through a memory interface. The caches are fast but have a small memory capacity. They are used to intermediately hold data that are used by the processor. The closer the cache is to a core's registers (the memory the computing units actually work on) the smaller and faster it usually is. [HW10, §1+§4]

A core requests needed data from the next higher level memory, i.e., the L1D cache. If the L1D does not hold the requested data, which is called a 'cache miss', the core must reach to higher levels of the memory hierarchy up to the comparatively slow main memory. In terms of performance it is therefore desirable to load data as fast and as early as possible and to use the loaded data as often as possible. This principle is the

basis for most of the performance optimizations presented in this thesis. In this context it also becomes important that different cores share distinct resources. Examples from figure 3.1b are the L3 cache and the memory bandwidth, which determines the maximum data transfer rate from the memory. [HW10, §1+4]

Each of the two processors in figure 3.1b has its own, local main memory that it accesses directly via its memory interface (local memory access). A single processor on this node together with its local memory is a uniform memory access (UMA) system because all cores on this single processor share the same route to the attached local memory. A core on the second processor can access the local memory of the first one through the IC network (remote memory access). Thus, the total memory attached to the node builds a single address space and the node is a shared memory computer [HW10, §4.2]. However, the route and access times for remote and local memory access are different, which makes this node a (typically cache-coherent) non-uniform memory access (ccNUMA) shared memory computer. The UMA building blocks therein are called NUMA domains. The whole cluster is therefore a distributed memory computer with shared memory ccNUMA nodes. [HW10, §4]

In the following I use the cluster setup as presented in this section since the UBT clusters btrzx5 and btrzx3 are of this type. Besides, there also exist, e.g., large scale shared memory computers [HW10, §4.2.3] or GPU clusters that rely on graphics cards. Furthermore, there is also progress in the field of high-performance computing and computer architectures. One example is the recent release of Intel's Xeon Phi Knights Landing processor [Sti16]. More details about computer architectures can be found in [HW10, §1+§4]. More information specifically about the UBT nodes can be found in appendix B.

3.2.2. Parallelization with MPI

To process a computation in parallel, the workload must be distributed among the cores of potentially many nodes. In principle there exist two ways of doing this:

First, functional parallelization divides a task into subtasks with different scopes. These are then processed by different programs, which communicate with each other to solve the complete problem. An example is the simulation of a car in a wind channel. One program simulates the car with a finite elements method while a second program simulates the wind by means of computational fluid dynamics.

Second, with data parallelization a single task is processed on a large set of data that is divided into subsets. A typical example is the division of the three dimensional (3D) space in a real-space code as BTDFT. There, each core (or process running on a core) gets a certain subgrid. The different processes must usually exchange data at the interfaces between neighboring subgrids. [HW10, §5.2]

There exist a lot of parallelization paradigms that are specialized in shared or distributed memory machines. A very common one is the message passing interface (MPI) [MPI12; Pac96], which is used within BTDFT. A program that is parallelized with MPI can be executed with a number of MPI processes that are mapped onto the hardware, i.e., the processor's cores. MPI processes on the same node can act on the same main memory. Still, each MPI process manages its own part of the main memory and therefore treats a physically shared memory as a logically distributed

one[5]. [HW10, §4.3]

During a computation the MPI processes communicate with each other by sending and receiving messages. Processes that communicate with each other are organized in communicators. Inside one communicator each process is identified by a unique rank. Still, each process can be part of several communicators and typically has different ranks in different communicators. [Pac96; MPI12]

In view of the communication between MPI processes it becomes important how the communicators are mapped onto the hardware. This can be controlled by the MPI execution command and a virtual process topology, which I discuss in a subsequent section. Sending a message between two processes takes a certain time, which is determined by the latency and the bandwidth of the respective network in between. In general, the communication takes longer the farther the processes are apart from each other. Typical situations are that the processes run, e.g., on the same processor, on different processors on the same node, or on different nodes in the communication network. The time for sending a message between two nodes across the network (i.e., the latter case) additionally depends on the relative position of the nodes in the network topology and how many switches the message must pass (compare to figure 3.1a). Latency and bandwidth are further explained in appendix C.1, supported by measurements on the btrzx5 cluster.

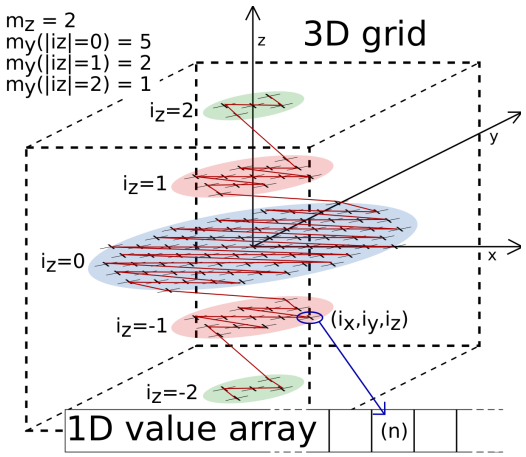
Finally, MPI for logically distributed memory computers is complemented by OpenMP [Ope13], which is designed for shared memory computers. In contrast to MPI, OpenMP only starts a single process that opens a number of threads in parallel regions that are specified inside the code. All threads share the same logical memory. OpenMP is not of particular importance in the current version of BTDFT but is used for some of the examples presented in the following sections. An example of OpenMP is given in appendix C.4.3 together with the pitfall of remote memory access when programming ccNUMA systems.

The concepts of MPI and OpenMP can be combined in a hybrid approach [HW10, §11]. To this end, one can start, e.g., one MPI process per NUMA domain with each process being parallelized with OpenMP. Each MPI process can then open threads that are pinned to the respective NUMA domain of the process. In this way the strengths of MPI, e.g., its scalability to large distributed memory computers, is combined with the strengths of OpenMP, which can partly make better use of shared resources.

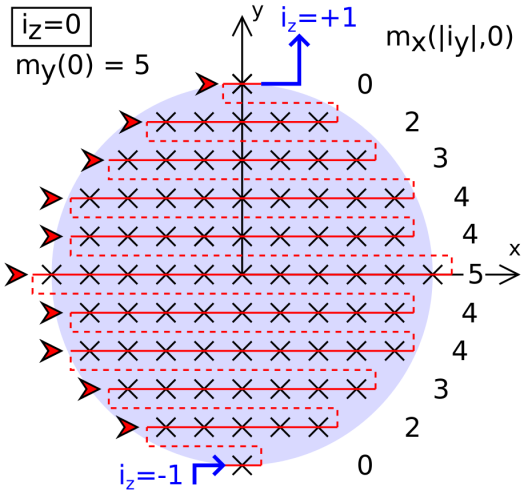
3.3. Grid setup, parallelization, and algorithms

In this section I introduce the basic setup of the real-space grid, the parallelization, and the most important algorithms that are use to solve the (TD)KS equations. In this course I lay special emphasis on the efficient application of semi-local operators such as the Laplacian or the Hamiltonian, which are the major performance bottlenecks. In appendix C I go deeper into the general performance engineering and discuss performance critical issues supported by measurements on the UBT clusters.

[5] With the MPI standard 3.0 [MPI12] shared memory windows were introduced.



(a) 3D grid stretched in z-direction. The five xy-planes are highlighted with different colors.



(b) xy-plane at $i_z = 0$.

Figure 3.2: Schematic of the BTDFT real-space grid. The contiguous 1D index is indicated by the red line (dashed segments in (b) connect adjacent x-rows). The 1D index runs through each x-row in the same direction as indicated by the red arrows in (b). Values for m_z, m_y, and m_x are given for this small example grid.

3.3.1. The real-space grid and grid parallelization

Grid setup The BTDFT real-space grid (see figure 3.2a) is centered around the origin of the coordinate system and framed by a boundary ellipsoid with the half-axes a_x, a_y, and a_z in x, y, and z-direction. The grid points are distributed equidistantly with a grid spacing of Δx in all three coordinate directions with one grid point at the origin. Their coordinates can therefore be described by a 3D index $\boldsymbol{i} = (i_x, i_y, i_z) \in \mathbb{N}^3$ through

$$\boldsymbol{r_i} = \boldsymbol{i}\Delta x \quad \text{with} \quad \sum_{\gamma=x,y,z} \frac{i_\gamma^2}{a_\gamma^2}(\Delta x)^2 < 1\,. \tag{3.1}$$

The total number of grid points is N_{grid}.

The maximum z-index (i_z) is called m_z. Therefore, i_z ranges within $-m_z \leq i_z \leq m_z$. The maximum y-index (i_y) in each xy-plane with z-index i_z is called $m_y(|i_z|)$. i_y runs within $-m_y(|i_z|) \leq i_y \leq m_y(|i_z|)$ in this xy-plane (see figure 3.2b). Finally, the maximum x-index (i_x) in each x-row with z-index i_z and y-index i_y is called $m_x(|i_y|, |i_z|)$. In this x-row, i_x is defined within $-m_x(|i_y|, |i_z|) \leq i_x \leq m_x(|i_y|, |i_z|)$. The whole grid shape can therefore be described by m_z, $m_y(|i_z|)$ and $m_x(|i_y|, |i_z|)$ with maximally possible indices of m_z, $m_y(0)$, and $m_x(0, 0)$ in x-, y-, and z-direction, respectively.

A function of space $f(\boldsymbol{r})$ is represented by its function values at the grid points $f(\boldsymbol{i}\Delta x)$. Due to the ellipsoidal shape of the grid the function values are stored in a one-dimensional (1D) value array with an 1D index $n = 1, \ldots, N_{\text{grid}} \in \mathbb{N}$, i.e., $f_n = f(\boldsymbol{i}\Delta x)$. In view of the later performance engineering (section 3.4) the one-to-one mapping between the 1D index n and the 3D index $\boldsymbol{i}$ is kept as simple as possible

(see figure 3.2): n always runs from the most negative index to the most positive one with the z-index being the most slowly varying one and the x-index being the fastest varying one. That means that n runs, one after another, through all xy-planes starting with the one at the smallest z-index. In each xy-plane, n runs, one after another, through all x-rows starting with the one at the smallest y-index. In each x-row, n again runs from the smallest to the largest x-index.[6]

Grid parallelization and halo communication The N_{grid} grid points are distributed evenly among a number of MPI processes such that each process gets a subgrid that is contiguous in the 1D index. This is exemplified in figure 3.3. Both subfigures show the decomposition of the grid into three subgrids, once from the perspective of the 1D value arrays and once for the real-space grid (as two-dimensional (2D) sketch). The white and blue highlighted grid points or array elements belong to a process's own subgrid.

The MPI processes frequently need information from each other and therefore send and receive messages. One of the most common situations where this happens is a spatial integral of some function that is defined on the real-space grid. In this case, the processes perform the integration locally in their own subgrids and subsequently sum up the local values using a collective MPI reduction.

BTDFT spends most of the computation time on the application of the Hamiltonian or the Laplacian, which are represented by finite differences of up to 6^{th} order ($N_{\text{order}} = 6$). In order to apply the finite-differences Laplacian to a function at a specific grid point, the function's values at the $N_{\text{order}}/2$ neighboring grid points are required in each of the six directions. This is shown in figure 3.3b. At the boundary of a process's subgrid there are boundary layers (blue highlighted in figure 3.3) that must be sent to the processes that contain the neighboring subgrids. Consequently, each process needs additional layers of grid points that store the function values received from processes with neighboring subgrids. These are called halo or ghost layers [HW10, §5.2.1] (red highlighted in figure 3.3) and must be updated each time before a finite differences operator is applied. The latter is called halo communication and indicated with green arrows in figure 3.3.

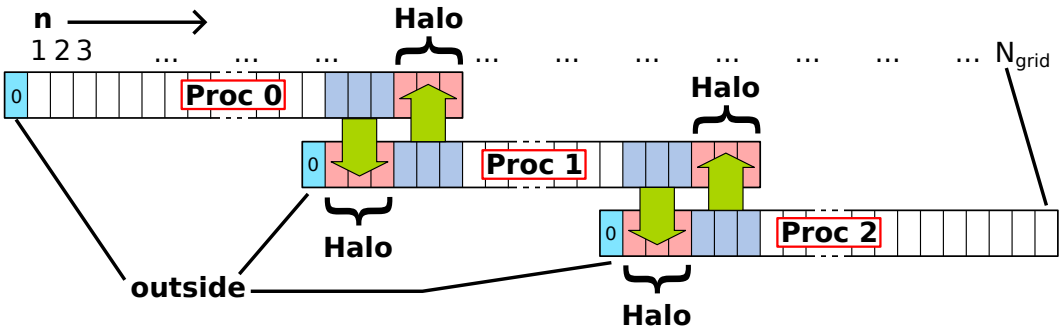
The grid structure chosen above ensures that the halo layers are almost[7] contiguous in the 1D index. Since BTDFT spends much computation time in the application of the Laplacian or the Hamiltonian, the halo layers are directly attached to all arrays in BTDFT that require halo communication. When performing local operations on those arrays, such as the multiplication with a number, the halo layers should, however, be excluded[8] since this would cause a lot of overhead.

In addition, figure 3.3a shows one additional array element at the lower end of the 1D arrays. This element contains the value of the respective function outside of the

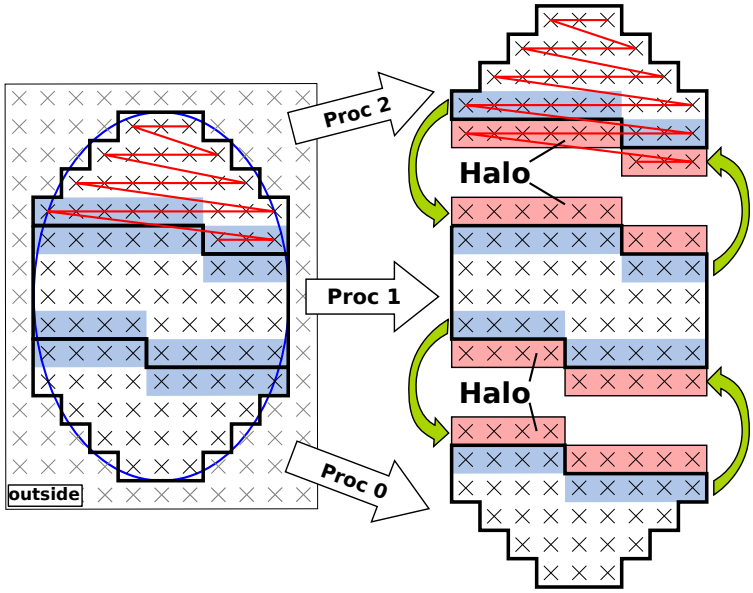
[6] In BTDFT the maps are stored in the index arrays *idx* (3D→1D) and *kx*, *ky*, and *kz* (1D→3D) in the t_grid structure.

[7] If the grid layers at the surface between two subgrids do not match perfectly, there appear single grid points that are not required by the respective neighboring process and do not need to be sent. Sending the pure halo layer without those values as a single message is still possible through derived MPI types and non-contiguous buffers [MPI12, §4]. Since this usually affects only a view single, isolated grid points in the boundary layers, this has no practical relevance.

[8] The process-specific 1D indices that define a process' own range and the halo layers are explained in appendix A.3.2 in more detail.



(a) 1D value arrays on the three processes.



(b) Decomposition of a 2D elliptical grid. The contiguous 1D index is indicated by the red line in the topmost subgrid.

Figure 3.3: BTDFT grid parallelization and halo communication among three processes. White and blue highlighted regions belong to a process's own subgrid, red highlighted regions are the halo layers that contain copies of the blue highlighted data from the respective neighboring process. The green arrows indicate the corresponding halo communication.

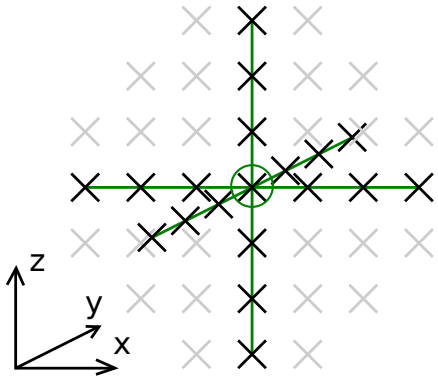


Figure 3.4: 6^{th} order finite differences stencil in three dimensions. The Laplacian is applied at the centered grid point. The function values at three neighboring grid points in each direction, connected by the greed line, are required.

grid and should contain 0 for the zero boundary conditions[9]. If the index array that converts a 3D index into its corresponding 1D index is applied to a 3D index outside of the grid, the resulting 1D index points to this element.

3.3.2. Orbital parallelization and virtual process topology

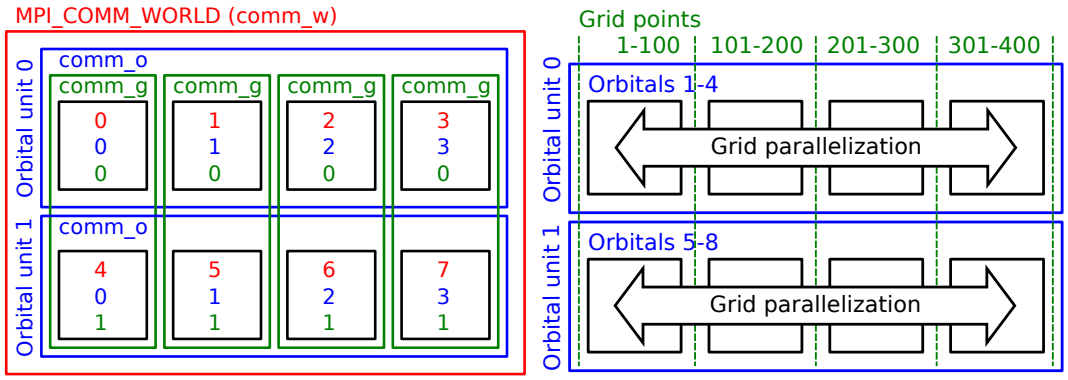
The solution of the TDKS equations in real time is divided into discrete time steps (see section 3.3.5). If the time-dependent potentials that appear in the TDKS equation 2.19 are known, the time step of a specific KS orbital can be done independently of the other KS orbitals. An additional orbital parallelization of the TDKS equation therefore makes sense, especially in view of large systems with many electrons.

To this end, the MPI processes are organized in orbital units with an equal number of processes per unit. Each orbital unit is responsible for a certain fraction of KS orbitals. This is illustrated in figure 3.5b for an example setup with eight processes, eight orbitals, and 400 grid points. The orbital units can be identified with the two rows in the graphic. Within the MPI processes of one orbital unit the computation is grid-parallelized as introduced in the previous section. In different orbital units there are processes that hold equal subgrid, just with different orbitals. These build the grid units, which are the columns in figure 3.5b.

Two layers of communication must be taken into account: The solution of the TDKS equation 2.19 requires the application of semi-local operators (see section 3.3.5) and therefore halo communication between the subgrids within the orbital units (intra-orbital-unit communication). After the time step the total density must be calculated (e.g., for new Hartree and xc potentials), which requires communication within the grid units, i.e., between the orbital units (inter-orbital-unit communication). As depicted in figure 3.5b this means that a certain process must communicate with its left and right horizontal neighbors as well as with all vertical neighbors.

When BTDFT is started with a certain number of MPI processes, they are organized in the default communicator MPI_COMM_WORLD (or $comm\_w$ in BTDFT).

[9] The solution of boundary value problems with non-zero boundary conditions at the surface of the grid is discussed in section 3.3.6.



(a) MPI communicators and ranks in corresponding colors.

(b) Example parallelization with 400 grid points and eight orbitals.

Figure 3.5: Virtual process topology with two orbital units and four processes each (black boxes). Processes within the same orbital unit (blue) share the same orbitals but have different subgrids, processes within the same grid unit (green) share the same subgrids but hold different orbitals.

The world communicator *comm_w* for the discussed example is indicated by the red box in figure 3.5a. The ranks of the processes in this communicator are given by the red numbers inside the processes (black boxes). In order to tell MPI as much as possible about the patterns of intra-orbital-unit and inter-orbital-unit communication, it is practical to reorganize the MPI processes in two additional communicators. The resulting orbital-unit (grid-unit) communicators *comm_o* (*comm_g*) are indicated by the blue (green) boxes with ranks in corresponding colors.

There is one master process that has extra tasks such as writing out the main status file or distributing BTDFT's configuration setup during the initialization. The global master process is the one with rank 0 in the world communicator *comm_w*. Additionally, each orbital unit has an orbital-unit master, which is the process with rank 0 in each orbital-unit communicator *comm_o*.

There exist different possibilities to map the virtual process topology onto the hardware. This can be controlled by the setup of the virtual topology and mapping options of the MPI command used to run BTDFT. The effect of different mappings is exemplified in appendix A.3.3 by means of a test systems from section 3.5. The default configuration is that orbital units are close-packed such that the more frequent intra-orbital-unit communication, i.e., exchanging the halo layers, is privileged. If a calculation is started, e.g., with as many orbital units as nodes, each node gets the processes that belong to one orbital unit.

There are some tasks between two time steps that are not orbital specific. Examples are the calculation of the Hartree potential or observables that only require the total density. The orbital parallelization does not work in this case and the respective task must be performed by each orbital unit separately. Another approach is to introduce an additional layer of the grid parallelization such that the non-orbital-specific operations can be performed grid-parallelized among all processes. This is explained in appendix A.3.2 together with some details of the grid setup inside of BTDFT.

The orbital parallelization introduces some additional overhead. One way to reduce

this overhead is to choose the time-step size Δt as large as possible. This reduces the total number of time steps and thus the inter-orbital-unit communication. A guideline on choosing the right parallelization and the number of orbital units can be found in appendix A.2.5 and in the performance tests in section 3.5.

3.3.3. Non-local pseudo potentials

The Coulomb potential $v_{\mathrm{Nuc}}(\boldsymbol{r}) = \sum_{i=1}^{N_{\mathrm{Nuc}}} \frac{-Z_i e^2}{4\pi\varepsilon_0 |\boldsymbol{r}-\boldsymbol{R_i}|}$ that is generated by N_{Nuc} nuclei with proton numbers Z_i shows singularities at the nuclei's positions $\boldsymbol{R_i}$. The core electrons that are bound to a nucleus have high kinetic energies and the KS orbitals show rapid spatial oscillations in this region. Hence, their proper description on the real-space grid would require a very fine grid spacing and thus large numerical effort. Yet, the chemical properties of a molecule are determined primarily by its valence electrons. [Kro+06]

To resolve these issues, BTDFT uses, in exactly the same way as PARSEC [Kro+06], norm conserving pseudo potentials from first principles for the description of the nuclei together with their core electrons. Pseudo potentials represent a screened, effective core potential. They are rather smooth in the proximity of the nuclei, which allows for a larger grid spacing and removes the core electrons from the explicit (TD)DFT calculation.

The remaining KS orbitals represent the valence electron density. Especially norm conserving Troullier-Martins pseudo potentials [TM91], which have been optimized for a plane-wave basis, are found to work very well on the real-space grid. However, the nuclei's pseudo potentials typically differ for components with different angular momentum (l, m) and require non-local projectors onto these components. The pseudo potential can be written as [Kro+06]

$$\hat{v}_{\mathrm{ps}}(\boldsymbol{r}) = \sum_{l,m} v_{\mathrm{ps},l}(\boldsymbol{r}) |l,m\rangle\langle l,m| \,. \tag{3.2}$$

The Kleinman-Bylander transformation [KB82] divides the pseudo potential into a local and a non-local part [Kro+06]

$$\hat{v}_{\mathrm{ps}}^{\mathrm{KB}}(\boldsymbol{r}) = v_{\mathrm{ps}}^{\mathrm{local}}(\boldsymbol{r}) + \hat{v}_{\mathrm{ps}}^{\mathrm{non-local}}(\boldsymbol{r}) \,. \tag{3.3}$$

The local part of the pseudo potential in equation (3.3) builds the local ionic potential in BTDFT. The non-local part includes the application of the angular-momentum projectors onto the KS orbitals, which requires a spatial integration in spheres around the nuclei with a certain cutoff radius [Kro+06]. The number of grid points inside of these spheres is usually small compared to the total number of grid points. Therefore, I neglect the non-local parts of the pseudo potentials in the performance considerations in section 3.4. From a numerical point of view the hermeticity and sparseness of the KS Hamiltonian is conserved such that the diagonalization of the KS equations and the propagation as introduced in the following is not influenced.

For the calculations presented in this thesis I used well established Troullier-Martins pseudo potentials [TM91], which are listed in appendix E.1. However, for magnesium a new pseudo potential had to be generated and tested, which is also discussed in appendix E.1.

3.3.4. Diagonalization of the Kohn-Sham equations

Since the KS potential (2.14) in the KS equation (2.7) depends on the density (2.9) that again depends on the KS orbitals, the KS equations must be solved self-consistently [KS65] within a self-consistent field (SCF) iteration. Beginning with an initial guess for the density, the KS potential can be calculated and the KS equation can be solved subsequently. This results in a new density that can be used in a further iteration. This procedure is repeated till convergence. The realization inside of BTDFT is explained in appendix A.3.1 by means of a program sequence.

The SCF scheme requires the frequent solution of the KS equation. The latter, which is a differential equation in space, can be cast into an eigenvalue problem through the spatial discretization on the real-space grid. This turns the KS orbitals into N_{grid}-dimensional vectors (the 1D value arrays) and the KS Hamiltonian, which essentially consists of the local KS potential and the semi-local Laplacian operator, into a sparse $N_{\text{grid}} \times N_{\text{grid}}$ matrix.

In BTDFT the eigenvalue problem is solved using the parallel version of ARPACK (PARPACK) [LSY97]. The latter only requires the frequent application of the KS Hamiltonian. Due to the exponential decay of bound orbitals outside a finite system [Kre+98], the KS orbitals are assumed to be zero at the grid's surface. Thus, zero boundary conditions apply.

As convergence criterion for the SCF iteration BTDFT uses the self-consistency residual error (SRE) [Kro+06]

$$SRE = \sqrt{\int [v_{\text{KS}}^{\text{new}}(\boldsymbol{r}) - v_{\text{KS}}^{\text{old}}(\boldsymbol{r})]^2 d^3r} \tag{3.4}$$

and the charge-weighted SRE

$$SRE_{\text{CW}} = \sqrt{\frac{\int [v_{\text{KS}}^{\text{new}}(\boldsymbol{r}) - v_{\text{KS}}^{\text{old}}(\boldsymbol{r})]^2 n(\boldsymbol{r}) d^3r}{\int n(\boldsymbol{r}) d^3r}} \,. \tag{3.5}$$

Both measure the difference between the KS potentials $v_{\text{KS}}^{\text{new}}$ and $v_{\text{KS}}^{\text{old}}$ between two consecutive SCF iterations. In addition I implemented their normalized counterparts

$$SRE^{\text{norm}} = \sqrt{\frac{\int [v_{\text{KS}}^{\text{new}}(\boldsymbol{r}) - v_{\text{KS}}^{\text{old}}(\boldsymbol{r})]^2 d^3r}{\int [v_{\text{KS}}^{\text{new}}(\boldsymbol{r})]^2 d^3r}} \tag{3.6}$$

and

$$SRE_{\text{CW}}^{\text{norm}} = \sqrt{\frac{\int [v_{\text{KS}}^{\text{new}}(\boldsymbol{r}) - v_{\text{KS}}^{\text{old}}(\boldsymbol{r})]^2 n(\boldsymbol{r}) d^3r}{\int [v_{\text{KS}}^{\text{new}}(\boldsymbol{r})]^2 n(\boldsymbol{r}) d^3r}} \,. \tag{3.7}$$

The charge-weighted variants give more weight to regions of space with higher density. Since these regions usually converge faster, the charge-weighted SREs lead to an overall faster convergence at the cost of typically less accurate KS potentials outside the electronic system.

The normalized SREs have the advantage of being bare numbers without a physical unit. While the non-normalized SREs measure a kind of absolute error of the total energy, the normalized SREs measure a relative error and hardly depend on the system

size. By default BTDFT uses the normalized SREs (charge weighted or not can be specified in the configuration file). The non-normalized SREs are computed as well but currently not used.

Finally, to stabilize the SCF scheme, KS potentials of previous iterations are usually mixed into the KS potential that is used for solution of the KS equation in the current iteration. Without this mixing, the SCF iteration can get stuck in a status in which it jumps back and forth between two states. This results in a constant SRE value and no further convergence. Two mixing algorithms are currently implemented in BTDFT: A linear one that mixes old and new potentials according to $av_{\mathrm{KS}}^{\mathrm{new}} + (1-a)v_{\mathrm{KS}}^{\mathrm{old}}$ with $a \in]0,1]$ and the Anderson mixer [And65] that uses an arbitrary number of previous potentials. References to different mixing schemes are listed in [Fio+03].

3.3.5. Propagation of the time-dependent Kohn-Sham equations

The real-time solution [SS96; YB96; YB99; CRS97; Cal+00; CMR04; MUN09; Mar03a; Cas+06; And+15] of the TDKS equation (2.19) is usually done by using the propagator [MUN09; CMR04]

$$\varphi_j(\boldsymbol{r},t) = U(t,t_0)\varphi_j(\boldsymbol{r},t_0) = \hat{T} exp\left\{-\frac{i}{\hbar}\int_{t_0}^{t} \hat{H}_{\mathrm{KS}}(\boldsymbol{r},t')\,dt'\right\}\varphi_j(\boldsymbol{r},t_0)\ . \tag{3.8}$$

$U(t,t_0)$ propagates the orbital φ_j from $t_0 \to t$. $\hat{T}$ is the time-ordering operator.

The propagator can be divided into time steps with size Δt according to

$$U(t,t_0) = U(t,t_0+(N_t-1)\Delta t)\ldots U(t_0+2\Delta t,t_0+\Delta t)U(t_0+\Delta t,t_0) \tag{3.9}$$

with $t = t_0 + N_t\Delta t$. If Δt is small enough[10], the KS potential can be assumed to be constant during one time step. Thus, the time-step propagator can be approximated by

$$U(t+\Delta t,t) \approx exp\left\{-\frac{i}{\hbar}\hat{H}_{\mathrm{KS}}\Delta t\right\}\ , \tag{3.10}$$

which represents a discretization of time. The KS Hamiltonian is usually taken at the time $t+\frac{\Delta t}{2}$, which requires a predictor-corrector scheme or a potential extrapolation [Pre+92]. Previous versions of BTDFT used the predictor-corrector method. However, the currently implemented polynomial extrapolation is more efficient and works at least as well. Since in the real-time propagation the time-step propagators are automatically applied in the correct causal order, the time ordering operator can be dropped. The time-step propagators are further approximated, which results, in the context of BTDFT, in the Taylor or Crank-Nicolson propagators described below[11].

Taylor propagator The Taylor propagator approximates the exponential in equation (3.10) by a Taylor series or fourth order

$$\varphi_j(\boldsymbol{r},t+\Delta t) \approx \sum_{n=0}^{4}\left[\frac{1}{n!}\left(-\frac{i\Delta t}{\hbar}\hat{H}_{\mathrm{KS}}\right)^n\right]\varphi_j(\boldsymbol{r},t)\,. \tag{3.11}$$

[10] This is either determined by the stability of the propagation algorithm or the time scales of the observed dynamics.

[11] See [MUN09; CMR04] for further propagators.

The fourth order Taylor propagator is the lowest order propagator that is conditionally stable and allows for reasonable time-step sizes. For each time step it requires four applications of the KS Hamiltonian. However, it is not norm conserving such that the orbitals are renormalized in BTDFT after every time step[12]. [MUN09; CMR04]

Crank-Nicolson propagator The Crank-Nicolson propagator is more evolved [Pre+92; Cal+00; CMR04]. It conserves the norm and the time-reversal symmetry per construction and is unconditionally stable. By using $U^{-1}(t,t') = U(t',t)$, the discretized propagation step is split up symmetrically such that $U(t+\frac{\Delta t}{2}, t+\Delta t)\varphi_j(t+\Delta t) = U(t+\frac{\Delta t}{2}, t)\varphi_j(t)$. The exponentials in the half-step propagators are then expanded in their Taylor series up to first order in Δt

$$\left[1 + \frac{i\Delta t}{2\hbar}\hat{H}_{\mathrm{KS}}\right]\varphi_j(\boldsymbol{r}, t+\Delta t) = \left[1 - \frac{i\Delta t}{2\hbar}\hat{H}_{\mathrm{KS}}\right]\varphi_j(\boldsymbol{r}, t)\,. \tag{3.12}$$

In view of the discretization of space equation 3.12 can be transformed into a matrix equation $\underline{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{b}$. The matrix $\underline{\boldsymbol{A}} = \left[1 + \frac{i\Delta t}{2\hbar}\hat{H}_{\mathrm{KS}}\right]$ is sparse, complex-valued, and symmetric. The right-hand side is defined as $\boldsymbol{b} = \left[1 - \frac{i\Delta t}{2\hbar}\hat{H}_{\mathrm{KS}}\right]\varphi_j(\boldsymbol{r}, t)$ and the solution vector on the left-hand side is defined as $\boldsymbol{x} = \varphi_j(\boldsymbol{r}, t+\Delta t)$.

In BTDFT the Crank-Nicolson equation (3.12) is solved using a variant of the iterative conjugate gradient (CG) method [SS07; Mei11] that works for complex-valued, symmetric matrices instead of hermitian ones. This variant, which I abbreviate by CG-SYM, can be derived from the more general bi-conjugate gradient (BiCG) [JAC86; Mei11] algorithm by assuming a symmetric (and in general complex) matrix[13]. As the standard CG algorithm, CG-SYM only requires the application of the matrix to a given vector (e.g., as a subroutine call).

As a convergence criterion I implemented the backward error [SS07]

$$\frac{\|\boldsymbol{b} - \underline{\boldsymbol{A}}\boldsymbol{x_n}\|}{\|\underline{\boldsymbol{A}}\|\,\|\boldsymbol{x_n}\| + \|\boldsymbol{b}\|} < \varepsilon_{\mathrm{tol}}\,, \tag{3.13}$$

where $\boldsymbol{x_n}$ is the approximate solution of the equation after n iterations and $\varepsilon_{\mathrm{tol}}$ is the convergence threshold or tolerance. In BTDFT the 2-norm is used as the vector norm and the matrix norm[14] $\|\underline{\boldsymbol{A}}\|$ is usually approximated by 0 or 1. BTDFT can automatically choose an optimized convergence threshold, which is explained in appendix A.3.4.

Due to its unconditional stability, the Crank-Nicolson propagator allows for much larger time steps than the Taylor propagator. However, every time step is computationally more expensive due to the solution of a linear equation that requires the

[12] This is omitted if absorbing boundary conditions are used [Dau16; Dau+16; Sch16; SK16]. The change of norm for typical time-step sizes is usually negligible.

[13] CG-SYM also works for local, complex-valued potentials on the diagonal of $\underline{\boldsymbol{A}}$ as they appear for, e.g., absorbing boundary conditions [Dau16; Dau+16; Sch16; SK16].

[14] In practice this is not relevant since the backward error does not specify which norm to use. However, a reasonable $\|\underline{\boldsymbol{A}}\|$ can be estimated by the spectral radius of $\underline{\boldsymbol{A}}$, which can again be estimated using the maximum kinetic single-particle energy that can be represented on the real-space grid. The result depends only on the grid spacing Δx and the order of the finite differences used.

frequent application of the operator $\underline{\boldsymbol{A}}$, which is essentially the Hamiltonian. The Taylor and Crank-Nicolson propagators are tested in section 3.5.1. In view of the orbital parallelization and the stability, the Crank-Nicolson propagator should typically be favored since a larger time-step size reduces the inter-orbital-unit communication.

3.3.6. Hartree potential and multipole expansion

Calculating the Hartree potential from the integral equation 2.15 is a heavy task on a 3D real-space grid. Alternatively, one can solve Poisson's equation [Bur+03; Kro+06]

$$\Delta v_{\mathrm{H}}(\boldsymbol{r}) = -\frac{e^2}{\varepsilon_0} n(\boldsymbol{r})\,. \tag{3.14}$$

This is a boundary value problem with zero boundary conditions at infinity. However, since the Coulomb interaction has a long range, the Hartree potential decays slowly such that typical grid sizes are by far not large enough that the Hartree potential can be assumed to be zero at the grid's surface.

To resolve this issue, one can divide the Hartree potential as $v_{\mathrm{H}} = v_{\mathrm{H}}^{\mathrm{asymptotic}} + v_{\mathrm{H}}^{\mathrm{residual}}$ into an asymptotic part and a residual. The asymptotic Hartree potential shall be exact outside the grid such that $v_{\mathrm{H}}^{\mathrm{residual}} = v_{\mathrm{H}} - v_{\mathrm{H}}^{\mathrm{asymptotic}}$ is zero outside the grid. Thus, Poisson's equation for $v_{\mathrm{H}}^{\mathrm{residual}}$ can be solved with zero boundary conditions at the grid's surface

$$\Delta v_{\mathrm{H}}^{\mathrm{residual}}(\boldsymbol{r}) = -\frac{e^2}{\varepsilon_0} n(\boldsymbol{r}) - \underbrace{\Delta v_{\mathrm{H}}^{\mathrm{asymptotic}}(\boldsymbol{r})}_{=-\frac{e^2}{\varepsilon_0} n^{\mathrm{boundary}}}\,. \tag{3.15}$$

n^{boundary} represents a density that generates $v_{\mathrm{H}}^{\mathrm{asymptotic}}$.

If the grid size is large enough, $v_{\mathrm{H}}^{\mathrm{asymptotic}}$ outside the grid is well described by a multipole expansion [Bur+03]. Still, there is some freedom in choosing the division of v_{H} into $v_{\mathrm{H}}^{\mathrm{asymptotic}}$ and $v_{\mathrm{H}}^{\mathrm{residual}}$ inside the grid.

In BTDFT (as well as in PARSEC [Kro+06]), $v_{\mathrm{H}}^{\mathrm{asymptotic}}$ is chosen to be zero inside the grid such that $v_{\mathrm{H}}^{\mathrm{residual}}$ is the full Hartree potential inside the grid. Outside the grid $v_{\mathrm{H}}^{\mathrm{asymptotic}}$ is set to the multipole expansion of usually ninth order. Therefore, $v_{\mathrm{H}}^{\mathrm{asymptotic}}$ and $v_{\mathrm{H}}^{\mathrm{residual}}$ are not steady at the grid's surface. With the discretized real-space grid and the Laplacian represented by finite differences, this is numerically not relevant. The boundary density $\Delta v_{\mathrm{H}}^{\mathrm{asymptotic}} = -\frac{e^2}{\varepsilon_0} n^{\mathrm{boundary}}$ inside the grid is non-zero only in a shell at the grid's surface at as many grid points as can be accessed by the finite differences from outside the grid.

Since a finite differences Laplacian on a real-space grid is represented by a real-valued, symmetric matrix, BTDFT solves equation 3.15 with the standard CG method [SS07; Mei11] using the backward error (3.13) as convergence criterion. On demand, the time-dependent code BTDFT_td can again choose an optimized convergence threshold, which is explained in A.3.4.

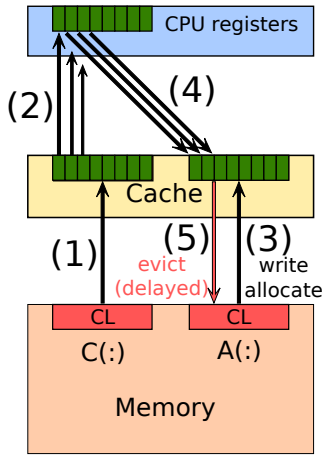


Figure 3.6: Memory access model for a copy operation [HW10, §3].

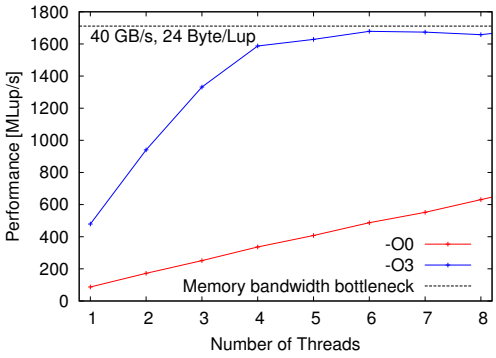


Figure 3.7: Parallel performance of the Jacobi smoother on one socket of the btrzx5 node258 (Intel Xeon E5-2670).

3.4. Performance engineering

The performance of a code depends on the hardware and how well the code can exploit the related resources. In the last decades a modern CPU's[15] theoretical potential in terms of operations per second evolved very rapidly while the speed for accessing the main memory could not keep up. Since many scientific codes such as BTDFT usually perform rather basic operations on very large data sets, a very common performance bottleneck is the memory bandwidth, i.e., the maximum rate at which data can be loaded from the main memory to a processor's cache. The memory bandwidth problem is particularly important for the application of the Hamiltonian, the Laplacian, or derivatives thereof. [HW10, §3.1]

In BTDFT the Hamiltonian operator is not stored as a matrix in the memory but implemented as a subroutine. Why the performance of BTDFT can be improved significantly by optimizing the memory access can be explained by the following example. In order to apply the discretized Laplacian as $g = \Delta f$ at a certain grid point, the values of f at the neighboring grid points in 3D space are required (see figure 3.4). Here, I want to emphasize that f exceeds the available cache by far. Thus, only small parts of f can be stored inside the cache at a time. A specific value of f, which corresponds to a specific grid point, is used several times since it is neighbor to several grid points. Therefore, this value should be kept inside the cache as long as it is needed to avoid loading it again from the main memory. If a value of f is still inside one of the cache levels when it is needed the next time strongly depends on the cache sizes, the size of the data set f, and the order in which f is stored and accessed. The latter depends on the mapping between the 3D grid and the 1D value arrays. This is elaborated in the following in more detail.

3.4.1. The memory-bandwidth bottleneck

The memory access is exemplified by a copy operation from one array C to another array A by means of a simple model with only one cache level [HW10, §3]. All steps discussed are displayed in figure 3.6. Therein one can see a memory that initially holds the array C, the cache, and the registers of the CPU[16]. In order to copy C, the CPU requests the data from the cache which again requests the data from the memory. The communication between the memory and the cache is done in units of cache lines, which are groups of typically 64 Byte, which equals 8 double precision (DP) words[17].

This means 8 DP words are loaded (step (1) in figure 3.6). Subsequently, the first value is loaded from the cache to the CPU's registers (step (2) in figure 3.6). In order to write the copied data to a cache line that corresponds to the array A, this cache line must also be loaded from the memory. This is called 'write allocate' (step (3) in figure 3.6). After that, the cache line of C can be copied to the corresponding cache line of A in the cache (step (4) in figure 3.6) and is finally written back to the memory, i.e., evicted (step (5) in figure 3.6) which can be delayed. In total the copy operation requires to load C, to store A, and an additional write allocate, which corresponds to loading A. As long as a cache line resides inside the cache, the data can be reused without loading them from the memory again, which opens the potential for memory-access optimizations.

The time for loading data from the memory can be characterized by the memory bandwidth and a latency time. The latter is more or less the time it takes to initiate the data transfer and has to be paid for every memory access (see appendix C.1). Organizing the data transfer in units of cache lines hides a part of the latency since it must only be paid once per cache line and not for every DP word. Furthermore, the processor tries to predict which data are needed in the near future and loads the data before they are actually required. This is called prefetching [HW10, §1.3.3]. In the following considerations the latency is assumed to be completely hidden by concepts as cache lines and prefetching. [HW10, §1+§3]

A single core of a modern processor cannot saturate the memory bandwidth. However, the same memory bandwidth is typically shared by many cores, which can easily put enough pressure onto the memory interface to hit the memory-bandwidth bottleneck. This is shown for a Jacobi smoother in listing 3.1 according to [HW10, §3.3].

The Jacobi smoother already has a similar structure as the 2^{nd} order finite difference Laplacian and is parallelized with OpenMP (see appendix C.4.3 for further comments). The parallel performance of the code, with and without compiler optimization, on one processor of the btrzx5 node258 (Intel Xeon E5-2670) (see appendix B.1) is shown in figure 3.7. The size of the data set has been chosen large enough that it does not fit completely into the cache. Thus, the code performance is bound to the memory bandwidth. All of the eight threads share the same memory interface. The performance is measured in Lup/s which means 'lattice site updates per second'. Hence, one Lup corresponds to the evaluation of the loop kernel in listing 3.1 for one grid point, i.e., one lattice site.

[15] A CPU in this thesis might be a processor or a single core.

[16] In the case of a copy operation the CPU is not performing any operation on the data, still the data must be loaded. This is why copy operations should be avoided.

[17] The cache-line size might vary between different CPUs and even between different cache levels.

```
do n=1,cycles
  !$omp parallel private(i,j,k)
  !$omp do
  do k = 1,nk
    do j = 1,nj
      do i = 1,ni
        y(i,j,k) = b * ( x(i-1,j,k) + x(i+1,j,k) + x(i,j-1,k) + &
                         x(i,j+1,k) + x(i,j,k-1) + x(i,j,k+1) )
      end do
    end do
  end do
  !$omp end do
  !$omp end parallel
  call dummy(x,y)
end do
```

Listing 3.1: 3D Jacobi smoother.

In figure 3.7 one clearly sees that the performance of the optimized code scales well for up to four threads and then saturates. The saturation limit can be understood if one relates the memory bandwidth with the size of the data that are loaded per Lup. The example shown in figure 3.7 has been chosen such that only one value of x must be loaded per Lup, which is the best possible case. In this best-case scenario, which is explained in section 3.4.3 in more detail, the remaining values of x are still inside the cache. Further, one value of y is stored and there is one write allocate. In total, this results in 3 DP words/Lup = 24 Byte/Lup, which is the same as for the copy operation discussed above.

The memory bandwidth for this specific processor is about 40 GB/s[18], which results in a maximum performance of about 1710 MLup/s. This performance marks the memory-bandwidth bottleneck in figure 3.7, which is almost approached by the code.

In contrast, the performance of the non-optimized code scales well for up to eight threads but does not get close to the bottleneck and therefore is simply not efficient. Since the data structure in BTDFT is, due to the ellipsoidal grid shape, more involved than the cubic one of the Jacobi smoother, the goal of the following performance engineering will be (i) to write code that is efficient enough to hit the memory-bandwidth bottleneck and (ii) to improve the memory-access patterns to make better use of every byte loaded from the memory, i.e., to raise the bottleneck as measured in MLup/s.

3.4.2. The BTDFT Hamiltonian

On a cubic grid the Hamiltonian with 6^{th} order finite differences would be implemented as in listing 3.2. The index and grid-shape variables are named as introduced in section 3.3.1 except that m_x and m_y are not arrays but single integers (due to the cubic grid shape). The array *orb* contains a complex-valued KS orbital with 16 Byte per double complex (DC) word and *horb* contains the Hamiltonian applied to the orbital. The real-valued potential *pot* with 8 Byte per DP word is added to the diagonal component

[18] The measured bandwidth is 41068.5 MB/s using the STREAM benchmark copy kernel [McC95] under consideration of the write allocate. The measured bandwidth in general differs between different kernels, i.e., a different ratio of load and store streams.

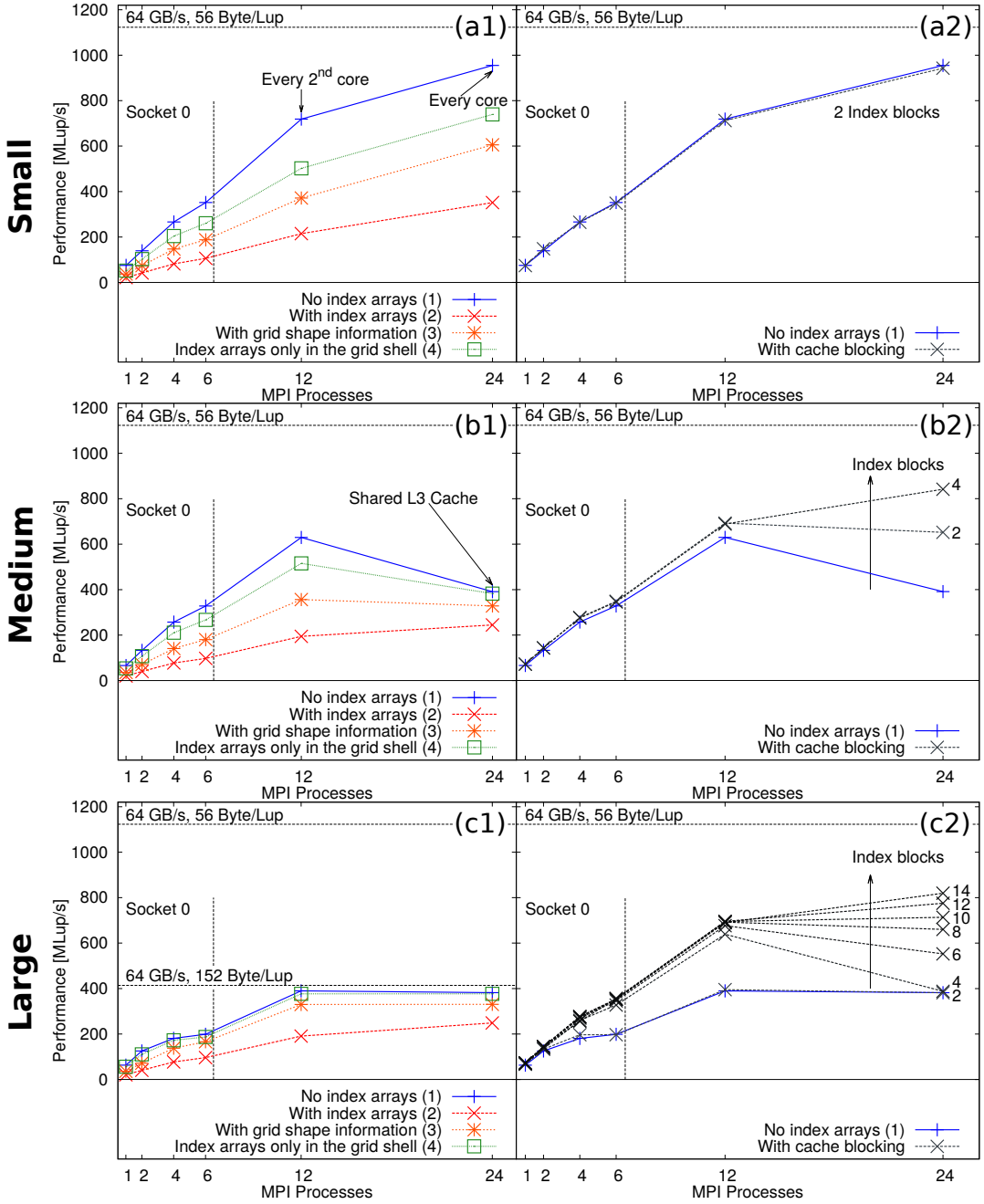


Figure 3.8: Performance of the 6^{th} order Hamiltonian with complex-valued orbitals depending on the number of MPI processes on one btrzx3 node (2 sockets) for three different system sizes and different numerical kernels described in the text. The MPI processes are scattered as far apart from each other as possible within one socket (up to 6 processes that use every second core) and then within the whole node (12 processes use every second core, 24 processes use every core). The halo communication is turned off such that the data only show memory-access effects.

```
do iz = -mz, mz
 do iy = -my, my
  do ix = -mx, mx
   horb(ix,iy,iz) = (pot(ix,iy,iz)+c(0))*orb(ix,iy,iz) +&
     c(1)*(&
       orb(ix-1,iy,iz) + orb(ix+1,iy,iz) +&
       orb(ix,iy-1,iz) + orb(ix,iy+1,iz) +&
       orb(ix,iy,iz-1) + orb(ix,iy,iz+1)) +&
     c(2)*(&
       orb(ix-2,iy,iz) + orb(ix+2,iy,iz) +&
       orb(ix,iy-2,iz) + orb(ix,iy+2,iz) +&
       orb(ix,iy,iz-2) + orb(ix,iy,iz+2)) +&
     c(3)*(&
       orb(ix-3,iy,iz) + orb(ix+3,iy,iz) +&
       orb(ix,iy-3,iz) + orb(ix,iy+3,iz) +&
       orb(ix,iy,iz-3) + orb(ix,iy,iz+3))
  end do
 end do
end do
```

Listing 3.2: Hamiltonian without index arrays.

of the Hamiltonian. The DP array $c(0:3)$ contains the constant coefficients of the 6^{th} order finite differences Laplacian for the central ($c(0)$) and the three neighboring ($c(1)$-$c(3)$) grid points on each side (see figure 3.4).

The performance of this code parallelized among a 2-socket btrzx3 node (2× AMD Opteron 6348 with 12 cores each) is shown in figure 3.8 (a1, Data set 1 'No index arrays') for a small grid for different numbers of MPI processes. The terms 'small', 'medium' and 'large' refer to the system size and are specified in section 3.4.3 in more detail. Up to six MPI processes are distributed among a single socket, i.e., one AMD processor on socket 0. They are scattered as far apart from each other as possible within this socket to make best use of the parallel resources. The calculations with 12 and 24 MPI processes ran on both sockets and correspond to 'every second core used' and 'every core used'.

The best-case memory-bandwidth bottleneck is again indicated as a black line at about 1120 MLup/s. The bandwidth of the node is approximately 64 GB[19]. The 56 Byte/Lup used for the bottleneck stem from the same memory-access model as for the Jacobi smoother from section 3.4.1 but with complex-valued orbitals and a real-valued potential. This means 1 DC + 1 DP = 24 Byte from loading one orbital value and one potential value, 1 DC = 16 Byte from storing the resulting value of *horb* and 1 DC = 16 Byte for the corresponding write allocate.

The performance of the code from listing 3.2, which is the performance reference in the following, scales well on the node and gets close to the memory-bandwidth bottleneck with 24 MPI processes. The halo communication has been turned off for these measurements such that only memory-access effects are measured.

[19] This has been measured with the LIKWID benchmark set with a result of 62869 MB/s. For the measurement I used the maximum number of OpenMP threads per NUMA domain and the copy kernel under consideration of the write allocate. The copy kernel describes the load and store streams only approximately in this case since three DP load streams come on two DP store streams due to the complex-valued orbitals.

```
do i = 1, ndim
 ix = kx(i)
 iy = ky(i)
 iz = kz(i)
 do order = 1,norder/2
  horb(i) = horb(i) + c(order)*(&
    orb(idx(ix-order,iy,iz)) + orb(idx(ix+order,iy,iz)) +&
    orb(idx(ix,iy-order,iz)) + orb(idx(ix,iy+order,iz)) +&
    orb(idx(ix,iy,iz-order)) + orb(idx(ix,iy,iz+order)))
 end do
end do
```

Listing 3.3: Hamiltonian with index arrays.

Due to the ellipsoidal grid shape of the BTDFT grid, the Hamiltonian cannot be implemented in the way displayed in listing 3.2 but one must include the mapping between the 3D grid index and the 1D index of the value arrays. The most simple way to apply the Hamiltonian in this case is shown in listing 3.3. In this piece of code there is no nested loop over the 3D grid but a single loop over the 1D value index. This is simpler because of BTDFT's grid parallelization that divides the 1D index range into contiguous parts.

The index arrays idx (3D $\rightarrow$ 1D) and kx, ky, and kz (1D $\rightarrow$ 3D), which are just conversion tables, hide the information about the grid shape, the mapping between the 3D grid points and 1D values and therefore the storage order of the value arrays in the memory. Moreover, the sum over the contributions from neighboring grid points with different distances is packed into an inner loop with the finite-difference coefficients in the array $c(0:3)$[20]. Still, if the code from listing 3.3 is applied to the same cubic grid as the previous one from listing 3.2 with the same finite-difference order, it does exactly the same computation.

The performance of the code from listing 3.3 is shown as data set 2 'With index arrays' in figure 3.8 (a1). It also scales with the number of MPI processes but its total performance does not reach the bottleneck. This has several reasons: First of all, the four index arrays, which posses as many elements as the 1D value arrays, have to be loaded additionally. This increases the bytes loaded per Lup and decreases the bottleneck performance. Second, the index arrays completely hide the grid shape and storage order of the value arrays in the memory from the compiler at the time of compilation and from the processor at run-time. The former prevents the compiler from optimizing the code, which alone has large impact on the performance (remember the non-optimized Jacobi smoother in figure 3.7). The latter enforces that a core must first load and evaluate an index array before it knows the next value that must be loaded. This prevents prefetching of data and therefore increases latency effects.

The question discussed in the following is how the code in listing 3.3 can be improved such that the performance gets closer to the one of the ideal code from listing 3.2, which is specialized for cubic grids. As a starting point one may think about the following

[20] In this implementation the order of the finite differences is not known at compile time and hence information hidden from the compiler. This alone already has huge influence on the performance and would also decrease the performance of the code in listing 3.2 by approximately 1/4 to 1/3 (not shown here). The effect is much smaller on the code from listing 3.3.

```
i = 0
do iz = -mz, mz
 do iy = -my(abs(iz)), my(abs(iz))
  do ix = -mx(abs(iy),abs(iz)), mx(abs(iy),abs(iz))
   i = i+1
   horb(i) = (pot(i)+c(0))*orb(i) +&
     c(1)*(&
       orb(idx(ix-1,iy,iz)) + orb(idx(ix+1,iy,iz)) +&
       orb(idx(ix,iy-1,iz)) + orb(idx(ix,iy+1,iz)) +&
       orb(idx(ix,iy,iz-1)) + orb(idx(ix,iy,iz+1))) +&
     c(2)*(&
       orb(idx(ix-2,iy,iz)) + orb(idx(ix+2,iy,iz)) +&
       orb(idx(ix,iy-2,iz)) + orb(idx(ix,iy+2,iz)) +&
       orb(idx(ix,iy,iz-2)) + orb(idx(ix,iy,iz+2))) +&
     c(3)*(&
       orb(idx(ix-3,iy,iz)) + orb(idx(ix+3,iy,iz)) +&
       orb(idx(ix,iy-3,iz)) + orb(idx(ix,iy+3,iz)) +&
       orb(idx(ix,iy,iz-3)) + orb(idx(ix,iy,iz+3)))
  end do
 end do
end do
```

Listing 3.4: Hamiltonian with grid information.

problem: Neither the compiler nor the process at run-time know the structure of the grid or the storage order of the 1D values arrays. However, this information is in principle available through the grid setup in section 3.3.1.

A first approach is therefore to use the grid-shape information that is given through the values of m_z and the arrays $m_y(:)$ and $m_x(:,:)$ from section 3.3.1 and the storage order of the value arrays in the sense that it is clear, per construction, how the 1D index runs through the 3D grid. One can start with the evaluation of the Hamiltonian at the 3D grid point that corresponds to the first 1D index in a process' range and run through the 3D grid in the same order as the 1D index. Thus, one only increments the 1D index in the innermost loop. In this way, the 1D index of the currently processed grid point as well as its 3D index are always known. Only the 1D indices of the neighboring grid points have to be looked up from the *idx* index array. The *kx*, *ky* and *kz* index arrays are replaced completely.

The corresponding Hamiltonian is shown in listing 3.4. Its performance is shown as data set 3 in figure 3.8 (a1)[21]. As expected, this code performs considerably better than that of listing 3.3. Still, there is room for further improvement.

Most of the remaining calls of the *idx* index arrays can be dropped if one additionally uses the fact that the 1D index runs through each x-row in the same direction (see figure 3.2). If the 1D index of the current grid point and of all its neighbors is known at the beginning of one row, one gets the 1D indices of the subsequent neighbors by incrementation. Therefore, the *idx* index array only has to be used inside a shell at the grid's surface with a width of as many neighbors as the finite differences scheme

[21] To ensure the comparability of the results, the performance measurement has again been done on the same cubic grid, i.e., the m_y and m_x arrays contain a single value. However, the compiler and the processor at run-time do not know that. Thus, the performance on the BTDFT grid can be simulated in this way.

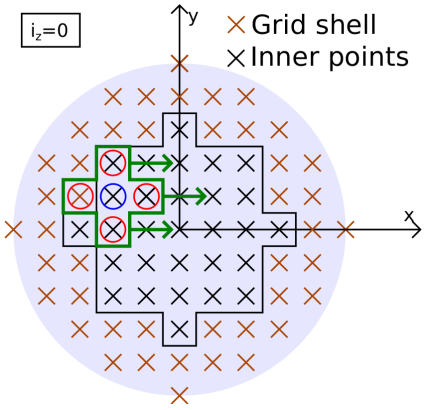


Figure 3.9: Finite differences of 2nd order in an xy-plane of the ellipsoidal BTDFT grid (compare to figure 3.2). The green stencil and arrows show the collective incrementation of the 1D indices of the currently processed grid point (blue circled) and the neighboring ones (red circled) in the inner region of the grid. The index array *idx* only needs to be used in the outer grid shell.

requires. This is shown graphically in figure 3.9.

The code from listing 3.4 with the additional improvements in the inner part of the grid as discussed by means of figure 3.9 is similar to the Hamiltonian as implemented in BTDFT. This implementation of the Hamiltonian (data set 4) increases the performance again towards the reference. The remaining difference is only due to the use if the *idx* array in the grid shell and the index bounds m_y and m_x, which are in general not single values. Thus, the performance should converge towards the reference if the system size is increased (to the 'medium' and 'large' systems).

The performance of the systems called 'medium' and 'large' is shown in figure 3.8 (b1) and (c1) for all of the implementations of the Hamiltonian discussed so far. For the medium system a sudden drop in performance appears at 24 processes for all versions including the reference. The performance of the large system is generally poor and at 12 processes already seems to hit a bottleneck that is far below the best-case bottleneck.

Still, the former expectation that the code from listing 3.4 with the improvements of figure 3.9 (data set (4)) converges towards the reference (data set (1)) holds true. This is not the case for the pure Hamiltonian from listing 3.4 without the improvements of figure 3.9 (data set (3)). The code corresponding to data set 4 is therefore reasonably efficient in the sense that it hits a bottleneck, especially for large systems that are the most critical ones in terms of total computation time. Just the bottleneck is not the one of the best case, which is discussed in the following.

3.4.3. Cache blocking

To estimate the memory bandwidth bottleneck in the former section, I used a best-case value of 56 Byte/Lup. This resulted from the assumption that only one value of the orbital array must be loaded per Lup. The performance drops for larger systems appear because more than one value has to be loaded.

This can be understood by a simple model: If a value has been loaded from the memory to the cache, it stays there until it is replaced due to the limited cache size. If that specific value is not required anymore by the Hamiltonian at the time it is replaced, the best case holds true. The values that are stored inside the cache are those that have been used last and are probably needed in the near future.

This is illustrated in figure 3.10 (a)-(c) by a 2D example with 2^{nd} order finite differences for the small, medium, and large systems. The arrows indicate the order the code runs through the 2D grid, which equals the storage order of the 1D index for figures 3.10 (a)-(c). The values that are inside the cache are highlighted[22]. This example appears in a similar way in [HW10, §3.3] for the Jacobi smoother above.

To evaluate the finite differences for the small system in this example, the top neighbor must in any case be loaded from the memory. The left neighbor has just been used and is for sure still inside the cache. When the top neighbor is loaded into the cache, the last value in the tail behind the currently processed grid point is evicted. If the cache is large enough to hold, in this example, at least two complete rows, the other two neighbors are also still inside the cache. This corresponds to best-case scenario from above (small system).

When the system's size gets larger in the horizontal direction such that less than two rows fit into the cache, one additional value must be loaded from the memory for each Lup (medium system). Thus, each value must be loaded twice in total. If the system gets even larger such that less than one row fits into the cache, also the third neighbor has to be loaded (large system). In total this means that more data have to be loaded per Lup and the performance at the memory-bandwidth bottleneck is decreased.

This example works analogously in three dimensions and with finite differences of higher order. For the 6^{th} order finite differences discussed above, in total six xy-planes must fit into the cache for the best-case memory access (small system). The bottleneck that is hit by the performance curves for the large system in figure 3.8 (c1) is simply that of the worst case.

The worst case happens if the cache holds less than a single xy-plane. In this situation the three neighbors in the xy-planes above and below as well as one value in the same xy-plane must be loaded[23]. This results in $7\,\text{DC} = 112\,\text{Byte}$ from loading the orbitals, $1\,\text{DP} = 8\,\text{Byte}$ from the potential, and $2\,\text{DC} = 32\,\text{Byte}$ for storing the resulting value and the associated write allocate. The resulting worst-case bottleneck with 152Byte/Lup is marked in the performance graph of the large system in figure 3.8 (c1) and fits perfectly to the data.

This example demonstrates that the code reacts sensitive on the cache size, which explains the performance drop of the medium system if the parallelization is increased from 12 to 24 MPI processes in figure 3.9 (b1). Since the L3 cache is shared among the processes, each of the 24 processes only gets half of the cache compared to running

[22] The real cache occupation is probably different. This is just an idealized example for illustration purposes.

[23] The values at neighboring grid points in x-direction are in the same x-row and have just been loaded into the cache. All values at neighboring grid points in y-direction, up to the foremost, have already been used in this xy-plane and are also still inside the cache. The next performance drop would appear when the cache gets too small to hold six x-rows which is not realistic in the context of BTDFT.

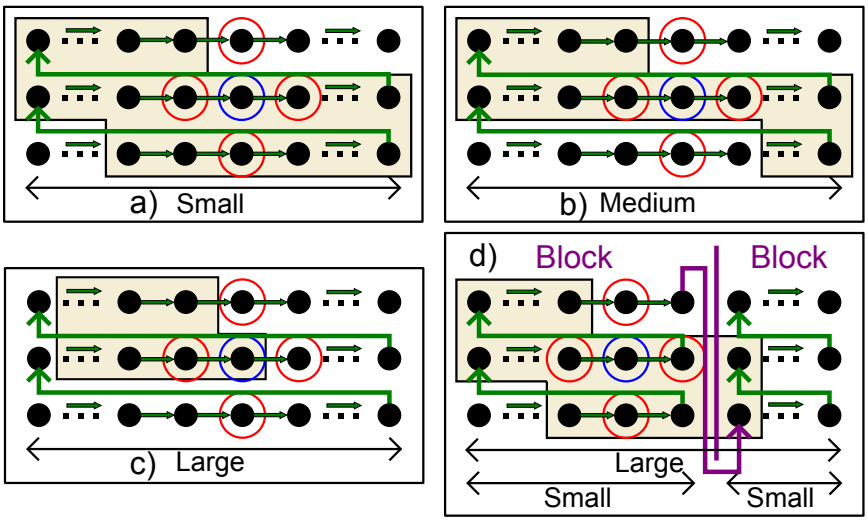


Figure 3.10: 2D example of the cache occupation with 2nd order finite differences for the small (a), medium (b), and large (c) systems and with additional cache blocking (d). The central blue-circled grid point is the one currently processed. The red-circled ones are those currently needed by the finite differences. The order the code runs through the grid is indicated by the green arrows. Data that are stored in the cache are highlighted.

the code with 12 processes. Therefore, one gets into the worst case with 24 processes and must load the data more often than in the case of 12 processes.

To counteract this effect, one can use cache blocking or index blocking [HW10, §3.5], which is illustrated in figure 3.10 (d). In the 2D example, one divides the grid into blocks by blocking the horizontal index. The Hamiltonian is then applied block-by-block. In a 3D grid is is usually sufficient to block one of the indices in the xy-plane. This adds an additional outermost loop. The mapping between the 1D index and the 3D grid is not influenced, only the order in which the Hamiltonian is applied changes.

In terms of the memory-access pattern, cache blocking has the effect of dividing a large system into many small ones. The overhead that is generated at the boundary between two blocks is usually negligible. In BTDFT the y-index is blocked as demonstrated in listing 3.5 by means of the reference code from listing 3.2. In this code the y-coordinate is divided into *nblock* blocks with start and stop y-index contained in the array *yblocks*.

The effect of cache blocking is shown in figure 3.8 (a2), (b2), and (c2) for the 3D reference Hamiltonian as discussed in a former section. As expected, the performance of the small system cannot be increased since it already shows the best-case memory access. However, the performance of the medium and large systems can be raised almost to the optimal case. The medium system already reacts on two and four blocks. The large system requires more blocks and the data set with only two blocks results in no improvement at all. For 24 MPI processes the performance can be enhanced significantly when, e.g., 14 blocks are used. In BTDFT the number of index blocks is determined automatically if the relevant cache size per process is given as preprocessor constant to the compiler (see appendix A.2.1).

```
do b = 1, nblock
 do iz = -mz, mz
  do iy = yblocks(b), yblocks(b+1)-1
   do ix = -mx, mx
    horb(ix,iy,iz) = (pot(ix,iy,iz)+c(0))*orb(ix,iy,iz) + &
    ![...]
   end do
  end do
 end do
end do
```

Listing 3.5: Hamiltonian without index arrays with blocked y-index.

3.5. Performance tests

To test the overall performance of BTDFT and to get a feeling for the performance for different time steps and the orbital parallelization, I discuss two cases: A linear polyacetylen chain with 80 KS orbitals and $5.2 \cdot 10^6$ grid points and a small aggregate of two bacteriochlorophylls *a* (BChl*a*) with 240 KS orbitals and $3.3 \cdot 10^6$ grid points. I ran the calculations of polyacetylen on the btrzx3 cluster and the BChl*a* calculations on the Intel Xeon E5620 nodes of the btrzx5 cluster (see appendix B) with various time steps and parallelizations for short propagation times. For the tests I always used the TDLDA functional, which can be evaluated with negligible effort between two time steps.

As a measure of performance I use the inverse of the computation time. However, I only show the relative performance (or scaling) with respect to a reference calculation. A scaling or relative performance of 'x2.00' means that this calculation requires half the computation time of the respective reference calculation, which translates to twice the performance.

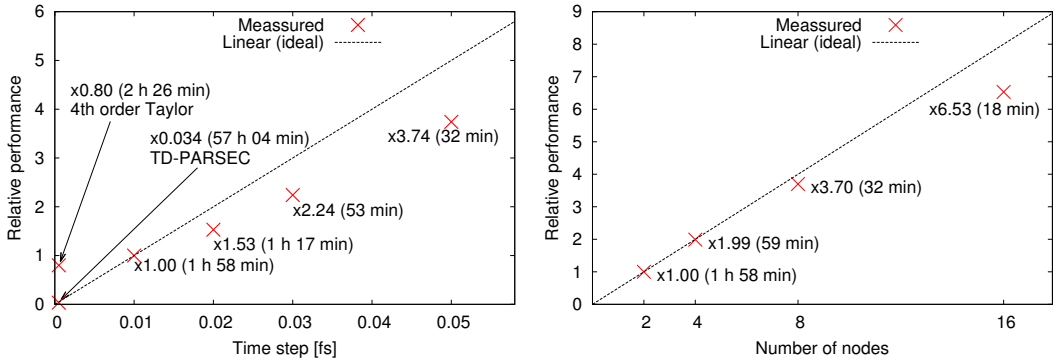
As already stated, there exist different possibilities to map the MPI processes onto the hardware. In the following the MPI processes in one orbital unit are close-packed. This specifically means that in a calculation with, e.g., 4 nodes and 2 orbital units, each orbital unit is distributed among two nodes. This setup privileges the halo communication, which happens more frequently. The effect of a mapping in which each orbital unit is scattered across all nodes is discussed in appendix A.3.3 and the scaling behaviour of BTDFT for the polyacetylen system is shown for up to 40 nodes on btrzx3. Numerical details for the presented calculations are listed in appendix E.2.1.

3.5.1. Polyacetylen chain

BTDFT was first planned as an extension of PARSEC [Kro+06] to time-dependent systems. However, in the Bayreuth version of PARSEC there already exists a time-dependent extension (TD-PARSEC), which uses the Taylor propagator. I use this as additional reference to make a statement about the absolute performance of BTDFT.

To this end, I first compare propagations on the polyacetylen system performed with

- TD-PARSEC with Taylor propagator for $\Delta t = 0.0005\,\mathrm{fs}$,



(a) Variation of the time step on two nodes of btrzx3 without orbital parallelization.

(b) Additional orbital parallelization ($\Delta t = 0.01\,\mathrm{fs}$). Two nodes share one orbital unit.

Figure 3.11: Performance of BTDFT on the polyacetylen system described in the text with respect to a reference calculation with only grid parallelization. The reference calculation (scaling factor 'x1.00') in both subfigures is the same with propagation time $T = 1\,\mathrm{fs}$ and time step $\Delta t = 0.01\,\mathrm{fs}$. Each orbital unit is grid-parallelized among two nodes of btrzx3. The dashed line marks the ideal scaling behaviour.

- BTDFT with Taylor propagator for $\Delta t = 0.0005\,\mathrm{fs}$, and
- BTDFT with Crank-Nicolson propagator for varying time-step sizes ranging from $\Delta t = 0.01\,\mathrm{fs}$ to $0.05\,\mathrm{fs}$.

The calculation is only grid-parallelized and the parallelization is kept constant among two nodes of the btrzx3 cluster with every second core occupied (i.e., 12 MPI processes per node). To make the calculations comparable, I aligned the linear polyacetylen along the parallelized Cartesian coordinate in TD-PARSEC and BTDFT such that halo communication is reduced to a minimum[24].

For the comparison between TD-PARSEC and BTDFT with the Taylor propagator, I used the same numerical parameters, as far as possible. As a result, for a propagation time of $T = 1\,\mathrm{fs}$ TD-PARSEC requires about 57 h whereas BTDFT requires only 2 h 26 min, which is a factor of > 23 faster.

Using the Crank-Nicolson propagator with larger time-step sizes can further increase the performance. The results are displayed in figure 3.11a, where the calculation with $\Delta t = 0.01\,\mathrm{fs}$ is chosen as reference. For comparison the performance from TD-PARSEC and BTDFT with the Taylor propagator are also depicted in figure 3.11a at $\Delta t = 0.0005\,\mathrm{fs}$ and indicated by arrows. The computation times given in the figure refer again to a propagation time of $T = 1\,\mathrm{fs}$.

The optimal scaling behaviour, which means a doubling in performance when doubling the time-step size, is indicated by the straight line in figure 3.11a. When the time-step size is increased, the performance increases as expected but does not scale optimally. Doubling the time-step size does not double the performance since the solution of the Crank-Nicolson equation requires more effort the larger the time step

[24] In TD-PARSEC this is the x-direction, in BTDFT this is the z-direction

is.[25]. Still, the Crank-Nicolson propagator improves the performance with respect to the Taylor propagator.

Finally, when the additional orbital parallelization is switched on, the number of nodes can be increased. The grid parallelization within each orbital unit is still kept fixed among two nodes such that the number of orbital units equals half the number of nodes. The resulting performance for different numbers of nodes (and hence orbital units) is shown in figure 3.11b with the Crank-Nicolson propagator and a constant time-step size of $\Delta t = 0.01\,\mathrm{fs}$. Hence, the reference value in figures 3.11a and 3.11b is the same.

One can clearly see that the orbital parallelization scales well. Even if eight times the resources are used (i.e., there are eight orbital units and in total 16 nodes), the performance is increased by a factor of 6.53. The latter is still close to the optimal scaling factor of 8.00[26].

3.5.2. Two bacteriochlorophylls

Since the polyacetylen system is a linear molecular chain, it requires only few KS orbitals but a relatively large grid. It is therefore not well suited for the orbital parallelization.

In addition I investigate an aggregate of two BChls with a smaller grid but three times the number of KS orbitals on the btrzx5 cluster[27]. I used this system to check the influence of different ratios between orbital parallelization and grid parallelization. The results of this test are shown in figure 3.12. Data points with the same color indicate an equal grid parallelization (but different total parallelization and number of orbital units). Data points with the same total parallelization but different numbers of orbital units, as indicated by the 'units' labels in figure 3.12, show different ratios between orbital parallelization and grid parallelization.

At this point I want to make three observations:

- For a total parallelization among four nodes the performance of the calculations with four orbital units is a factor of about 1.72 faster than the one with one orbital unit (i.e., only grid parallelization among four nodes). This means that a grid parallelization among four nodes is not efficient for this example.

- For a parallelization among eight nodes the same can be stated since the calculation with two orbital units (i.e., again a grid parallelization among four nodes) is slower than the calculations with four and eight orbital units (i.e., a grid parallelization among two nodes or one node, respectively.). However, the performance for four and eight orbital units is quite the same, which means that a grid parallelization among two nodes is still efficient.

- Keeping the grid parallelization among two nodes but doubling the total parallelization to 16 nodes with eight orbital units almost doubles the resulting

[25] The larger a time step is, the larger is typically the difference between the time-dependent potentials at t and $t + \Delta t$.

[26] The approximately constant time for initialization and output of about 1 min disturbs the result a little for these small total computation times. If this is taken into account, the scaling factor for 16 nodes is about 6.85.

[27] Nodes with two Intel Xeon E5620 processors, eight MPI processes per node, see appendix B.1.

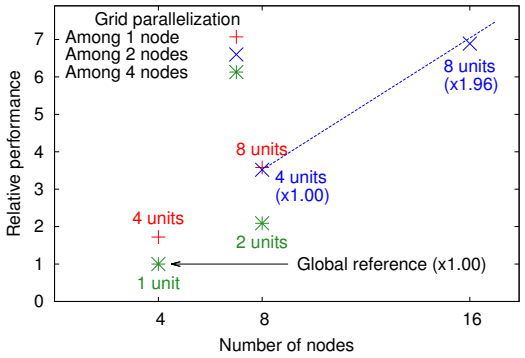


Figure 3.12: Performance of the BChl*a* aggregate on btrzx5 with different parallelizations. The data sets with equal colors have the same grid parallelization. For the blue data set with a grid parallelization among two nodes the ideal scaling curve is shown.

performance (factor of 1.96 compared to the optimal scaling factor of 2.00). The optimal scaling behaviour with the calculation with eight nodes and four orbital units as reference is indicated by the straight line.

In summary, the grid parallelization is already maxed out for two (or maybe three) nodes and further grid parallelization is not efficient for the BChl*a* system. However, by using additional orbital units, the parallelizability is enhanced by far. The almost perfect scaling from eight to 16 nodes with the same grid parallelization (blue data points) indicates that the code scales also well for $\gg 16$ nodes and underlines the efficiency of the orbital parallelization. Thus, one can conclude that BTDFT is expected to be well suited for TDDFT simulations of the large multichromophoric systems that are the subject of subsequent sections and future work.

4. Evaluation of electronic spectra and transition densities

The energy transfer in light-harvesting systems such as the LH2 complex, which is introduced in the next section, is mediated through excited electronic states. For its correct theoretical description by means of real-time TDDFT, the proper description of the participating excited states is mandatory.

Electronic excitations can be described by their energy, oscillator strength (or dipole strength), and transition density [KAR01; Bro+04; TK09]. Typical quantum chemistry codes, such as Gaussian [Fri+] or QChem [Sha+15], solve the TDKS equations in frequency space within the linear-response formalism [CAS95; PGG96; GDP96; Cas+98] and return these quantities quite naturally. In contrast to that, the real-time implementation [SS96; YB96; YB99; CRS97; Cal+00; CMR04; MUN09; Mar03a; Cas+06; And+15] requires more effort to extract those data since its only output is the time-dependent density $n(\boldsymbol{r}, t)$.

Optical excitations are usually extracted from the time-dependent dipole moment or, more exactly, its Fourier transform after a dipole-like excitation as shown below [SS96; YB96; CRS97; YB99]. However, the direct interpretation of the results as optical spectra limits the accuracy due to the finite resolution of the discrete Fourier transform and the finite propagation time [CRS97]. The quantitative information about single excitation energies and oscillator strengths can still be extracted from those spectra by fitting excitations with the correct line shape. This was already pointed out in [CRS97].

Predicting accurate excitation energies from real-time TDDFT is still of interest [And+15]. Therefore, I advanced the existing evaluation scheme for dipole active electronic excitations and their transition densities from real-time TDDFT and present the results in this section. This technique is fundamental for the discussion in section 5 since it returns, within the approximations done in applied TDDFT, accurate values for excitation energies and oscillator strengths. It even reveals small excitations that are otherwise overseen. The same is true for the transition densities, which directly display the motion of the density at a certain excitation frequency[28] and characterize the respective transition. The technique presented here makes transition densities of weak transitions accessible from real-time TDDFT and even useful for quantitative purposes. This is applied to the calculation of coupling strengths between chromophores in section 5.5.

4.1. Traditional real-time evaluation of spectra and transition densities

In order to calculate the photoabsorption spectrum from real-time TDDFT, one initially applies a boost excitation to the ground-state KS orbitals $\varphi_j^{(\mathrm{GS})}$ [YB96]

$$\varphi_j(\boldsymbol{r}, t = 0) = e^{i\boldsymbol{k}\cdot\boldsymbol{r}} \varphi_j^{(\mathrm{GS})}(\boldsymbol{r})\,. \tag{4.1}$$

If the boost strength $k = |\boldsymbol{k}|$ is small, this corresponds to a dipole-like, instantaneous kick with energy $E_{\mathrm{boost}} = N\frac{\hbar^2 k^2}{2m}$ and direction of $\boldsymbol{k}$. N is the number of electrons, m

[28] The terms excitation frequency and excitation energy are used equivalently in this work due to their direct correspondence $E_{\mathrm{excit}} = \hbar\omega_{\mathrm{excit}}$.

is the electron mass, and $\hbar$ is Plank's constant. The meaning of the boost is explained below in more detail.

The induced dipole moment

$$\boldsymbol{\delta\mu}(t) = \boldsymbol{\mu}(t) - \boldsymbol{\mu}^{(\mathrm{GS})} \tag{4.2}$$

with

$$\boldsymbol{\mu}(t) = -e \int \boldsymbol{r} n(\boldsymbol{r}, t) d^3 r \tag{4.3}$$

displays the density oscillations of dipole active excitations with their corresponding excitation energies. e is the elementary charge.

If the system is excited in three orthogonal directions (here: $\vartheta \in \{x, y, z\}$) in three separate calculations, this results in three induced dipole moment vectors $\boldsymbol{\delta\mu}^{(\vartheta)}$. The componentwise ($\gamma \in \{x, y, z\}$) Fourier transform of the three dipole vectors builds the dynamic polarizability tensor [YB99; Mar03a]

$$\alpha_{\vartheta\gamma}(\omega) = \frac{-e}{\hbar k} \mathcal{F}\left[\delta\mu_\gamma^{(\vartheta)}\right](\omega)\,. \tag{4.4}$$

The Fourier transform $\mathcal{F}$ from the time domain to the domain of angular frequency is in the following denoted by a tilde, i.e., $\tilde{f} = \mathcal{F}[f]$. The imaginary part of the trace of the polarizability tensor corresponds directly to the photoabsorption cross section (in CGS units) [ZS80; Mar+12; Mar03a]

$$\sigma(\omega) \overset{\mathrm{CGS}}{=} \frac{4\pi\omega}{3c} \Im\left\{\mathrm{Tr}\left[\alpha_{\vartheta\gamma}(\omega)\right]\right\}\,. \tag{4.5}$$

The imaginary part is denoted by $\Im$ whereas the trace operator is Tr and c is the speed of light. Building the trace in the photoabsorption cross section corresponds to averaging over all orientations of the system. This removes the dependence of the resulting spectrum on the direction of the boost excitation, which becomes clear in the derivation below.

An equivalent quantity, which is commonly used, is the dipole strength function (DSF) [ZS80; YB99; Mar+12]

$$S(\hbar\omega) = -\frac{2m\omega}{3\pi e\hbar^2 k} \Im\left\{\mathrm{Tr}\left[\widetilde{\delta\mu}_\gamma^{(\vartheta)}(\omega)\right]\right\} \tag{4.6}$$

The DSF integrates to the total oscillator strength, which must equal the number of electrons due to the Thomas-Reiche-Kuhn or f-sum rule [YB96; YB98; YB99; Mar03a].

The absorption cross section and the DSF show peaks at the excitation energies of dipole active singlet excitations. These peaks are artificially broadened due to the finite propagation time T of the real-time simulation, which is explained below in more detail. Usually, the outcome of equations (4.5) or (4.6) is directly compared to optical spectra from the experiment or excitation energies read out from the peak positions. [YB96; CRS97; YB99; Cal+00; LSR02; MG04; MK06; MK07; WU08; MUN09; TK09; HKK12; HK12; Mar+12]

In this context, the power spectral density (PSD) [CRS97; MUN09; Pre+92] of the time dependent dipole moment

$$P(\hbar\omega) = 2\,\mathrm{Tr}\left[\left|\widetilde{\delta\mu}_\gamma^{(\vartheta)}(\omega)\right|^2\right] \tag{4.7}$$

is another commonly used quantity[29]. If one is only interested in the excitation energies of strong dipole excitations, it is usually sufficient to excite the system once with a boost that is non-orthogonal to the dipole excitations of interest.

The DSF and the PSD of the Na_4 cluster and a BChl$a$[30] are drawn in figure 4.1 (a1)+(a2) and (b1)+(b2). The spectra arise from real-time propagations with a propagation time of $T = 50\,\mathrm{fs}$ (Na_4) and $T = 100\,\mathrm{fs}$ (BChl$a$), respectively. They are typically found this way in the literature [YB96; CRS97; YB99; Cal+00; LSR02; MG04; MK06; MK07; WU08; MUN09; TK09; HKK12; HK12; Mar+12]. For both systems the spectra only display the energy range up to about 3.5 eV and 2.1 eV, respectively. The discrete Fourier transform was done with the Fast Fourier Transform (FFT) algorithm [Pre+92]. Further numerical details for the calculations presented in this section are listed in appendix E.2.2.

DSF as well as PSD for both molecules show, as expected, peaks at excitation energies. For Na_4 three to five excitations can be identified from figures 4.1 (a1)+(a2). For BChl$a$ two peaks can be identified from figures 4.1 (b1)+(b2). The latter correspond to the prominent Q-band excitations [Gou61; Hu+02; CGK06] Q_y (larger one) and Q_x (smaller one) excitations (see section 5). Yet, the spectra look quite spiky and the identification and evaluation of excitations is questionable. Smaller excitations that are close to the identified larger ones can be overseen and excitation energies can only be determined with a resolution of $\Delta\omega = \frac{2\pi}{T}$, which is the sampling rate in the frequency domain from the discrete Fourier transform [Pre+92].

Moreover, the DSF shows negative values, which is incompatible with its direct interpretation as an absorption cross section. One possibility to resolve this issue is to damp the time-dependent dipole moment, e.g., with a decaying exponential function, which simulates a kind of finite lifetime of the excited states. If the signal is damped sufficiently, this results in smooth spectral lines but is accompanied by a loss of spectral resolution. [YB96; CRS97]

Finally, the transition densities, which directly display the density oscillation at a given excitation energy, suffer from similar problems. From a real-time simulation they are usually evaluated as the Fourier transform of the induced density fluctuation $\delta n(\boldsymbol{r}, t) = n(\boldsymbol{r}, t) - n^{(\mathrm{GS})}(\boldsymbol{r})$ [KAR01; TK09; HKK12; HK12]

$$\rho_{if}(\boldsymbol{r}) = \langle i|\hat{n}(\boldsymbol{r})|f\rangle \propto \Im\{\tilde{n}(\boldsymbol{r}, \omega_{if})\} \tag{4.8}$$

for the excitation from an initial state the $|i\rangle$ to a final state $|f\rangle$ with energy $\hbar\omega_{if}$ and the density operator $\hat{n}$.

4.2. Refined excitation energies and oscillator strengths

For the later discussion detailed information about small, artificial states is required. In the following, I elaborate a scheme to evaluate the dipole spectra for accurate excitation energies, oscillator strengths, and transition densities. To this end, I first derive the expression for the DSF after a numerical propagation with a weak boost.

[29] I define the PSD with a factor of 2, which is explained in [Pre+92] (one-sided PSD).
[30] Structure taken from [ONS10].

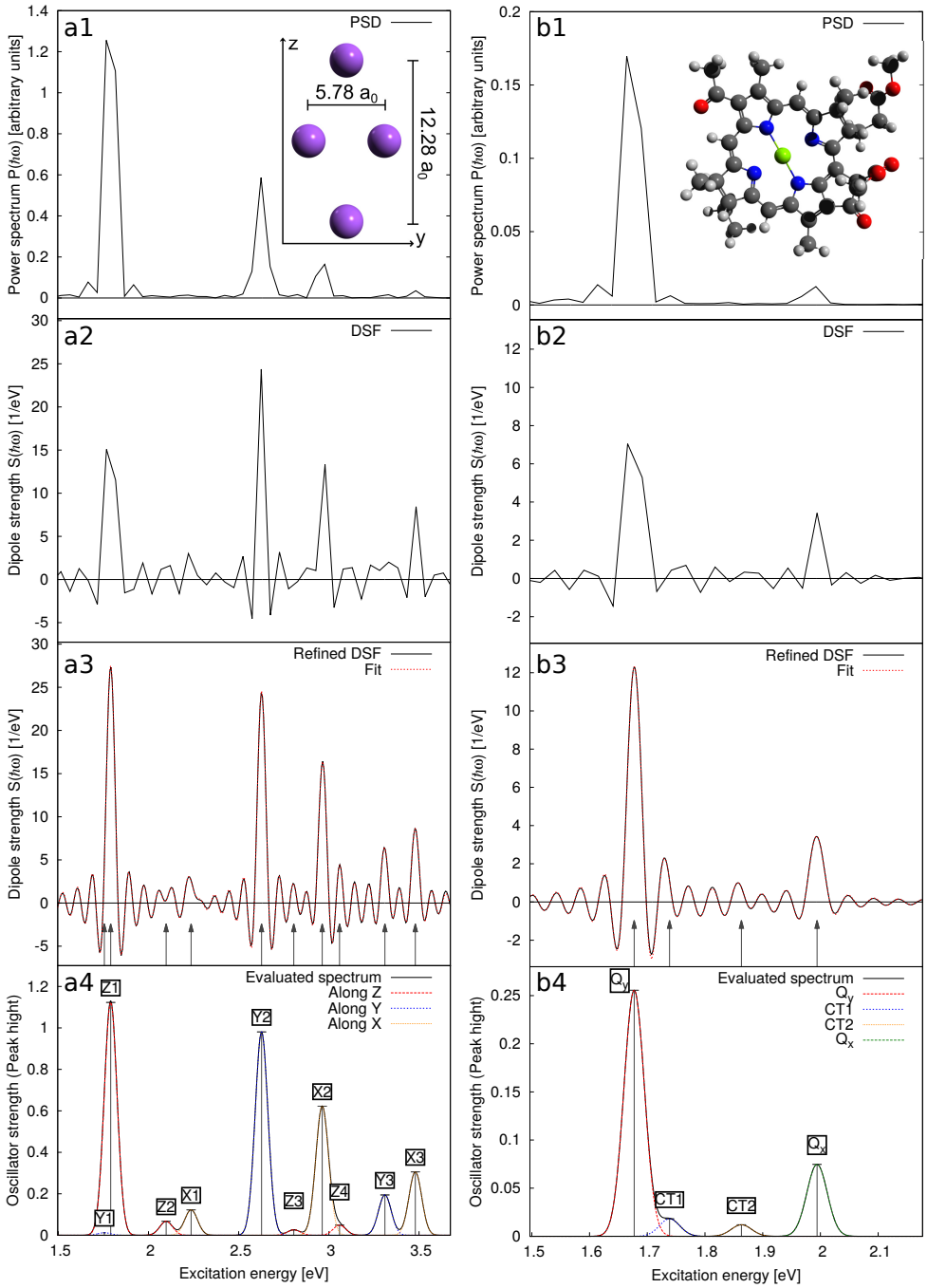


Figure 4.1: PSD (a1)+(b1), DSF (a2)+(b2), refined DSF with fitted excitations (a3)+(b3), and the evaluated spectrum (a4)+(b4) of Na_4 ($T = 50\,\mathrm{fs}$) (a1)-(a4) and BChl$a$ ($T = 100\,\mathrm{fs}$) (b1)-(b4). The lines in the evaluated spectrum are artificially broadened by Gaussian functions $e^{-[\hbar(\omega-\omega_{0j})/\eta]^2}$ with $\eta = 0.05\,\mathrm{eV}$ (a4) and $\eta = 0.025\,\mathrm{eV}$ (b4).

The time-dependent dipole moment The real-valued many-body eigenstates of the ground-state Hamiltonian are denoted by $\{|j\rangle\}$ with the ground state $|0\rangle$ and energies $E_j = \hbar\omega_j$. The system is in its ground state for $t < 0$ and is subject to a boost excitation at $t = 0$. The many-electron wave function at $t = 0$ reads

$$|\psi(t=0)\rangle = e^{i\boldsymbol{k}\cdot\hat{\boldsymbol{r}}}\,|0\rangle\ . \tag{4.9}$$

The boost acts on all electronic coordinates as $e^{i\boldsymbol{k}\cdot\sum_{j=1}^{N}\boldsymbol{r}_j}$. The time-dependent wave function can be expanded into the basis $\{|j\rangle\}$

$$|\psi(t)\rangle = \sum_{j=0}^{\infty} c_j\,|j\rangle\,e^{-i\omega_j t}\,. \tag{4.10}$$

The expansion coefficients c_j are constant during the evolution for $t > 0$ since the Hamiltonian is time-independent after the boost. The coefficients are determined by the initial condition

$$c_j = \langle j\,|\psi(t=0)\rangle = \langle j|\,e^{i\boldsymbol{k}\cdot\hat{\boldsymbol{r}}}\,|0\rangle\ . \tag{4.11}$$

The time-dependent dipole moment follows from

$$\boldsymbol{\mu}(t) = \langle\psi(t)|\,-e\hat{\boldsymbol{r}}\,|\psi(t)\rangle = -e\sum_{j,l=0}^{\infty} c_l^* c_j e^{-i\omega_{lj}t}\boldsymbol{r}_{lj}\,. \tag{4.12}$$

The asterisk denotes complex conjugation. Further, $\omega_{lj} = \omega_j - \omega_l$ is the excitation frequency for the $l \to j$ transition and $\boldsymbol{r}_{lj} = \langle l|\,\hat{\boldsymbol{r}}\,|j\rangle$ the corresponding transition dipole moment [31].

For small boost strengths the exponential in equation (4.11) can be expanded into a finite Taylor series. This way, the expansion coefficients reduce to

$$c_j \approx \langle j|\,1 + i\boldsymbol{k}\cdot\hat{\boldsymbol{r}}\,|0\rangle = \delta_{j0} + i\boldsymbol{k}\cdot\boldsymbol{r}_{j0}\,. \tag{4.13}$$

The weak boost acts as a dipole-like excitation with the direction of $\boldsymbol{k}$.

By neglecting terms proportional to $(\boldsymbol{k}\cdot\boldsymbol{r}_{j0})(\boldsymbol{k}\cdot\boldsymbol{r}_{l0})$, the products $c_l^* c_j$ in equation (4.12) read

$$c_l^* c_j \approx \delta_{l0}\delta_{j0} + i\boldsymbol{k}\cdot(\delta_{l0}\boldsymbol{r}_{j0} - \delta_{j0}\boldsymbol{r}_{l0})\ . \tag{4.14}$$

The first term in this equation leads to the static dipole moment of the ground state $\boldsymbol{\mu}(t=0) = -e\langle 0|\hat{\boldsymbol{r}}|0\rangle$, which is of no interest. The remaining induced dipole moment is

$$\begin{aligned}
\boldsymbol{\delta\mu}(t) &\approx -ie\sum_{j,l=0}^{\infty} \boldsymbol{k}\cdot(\delta_{l0}\boldsymbol{r}_{j0} - \delta_{j0}\boldsymbol{r}_{l0})\,e^{-i\omega_{lj}t}\boldsymbol{r}_{lj} \\
&= -ie\sum_{j=0}^{\infty} (\boldsymbol{k}\cdot\boldsymbol{r}_{0j})\,\boldsymbol{r}_{0j}\left[e^{-i\omega_{0j}t} - e^{i\omega_{0j}t}\right] \\
&= -2e\sum_{j=1}^{\infty} (\boldsymbol{k}\cdot\boldsymbol{r}_{0j})\,\boldsymbol{r}_{0j}\sin(\omega_{0j}t)\,.
\end{aligned} \tag{4.15}$$

[31] The transition dipoles are $-e\boldsymbol{r}_{lj}$. However, I also use the term freely without the factor $-e$.

Here, $\omega_{jl} = -\omega_{lj}$ and $\boldsymbol{r}_{j0} = \boldsymbol{r}_{0j} \in \mathbb{R}$ have been used. The induced dipole moment oscillates, as expected, with the excitation frequencies ω_{0j} of the $0 \to j$ transitions. The amplitudes of the single spectral components depend only on their respective transition dipoles and on their angle with respect to the boost vector through the inner product $\boldsymbol{k} \cdot \boldsymbol{r}_{0j}$.

The dependence on the angles between boost and transition dipoles can be removed by performing three calculations with orthogonal boost vectors $\boldsymbol{k}^{(\vartheta)} = k\hat{\boldsymbol{e}}_\vartheta$. In the following, I use the Cartesian directions with $\vartheta \in \{x, y, z\}$, where $\hat{\boldsymbol{e}}_\vartheta$ are the corresponding unit vectors. The trace of the dipole moment matrix $\delta\mu_\gamma^{(\vartheta)}$ with $\gamma \in \{x, y, z\}$ reads

$$\mathrm{Tr}\left[\delta\mu_\gamma^{(\vartheta)}\right] = \sum_{l=x,y,z} \delta\mu_l^{(l)} = -2ek \sum_{j=1}^{\infty} |\boldsymbol{r}_{0j}|^2 \sin(\omega_{0j}t) = -\frac{3e\hbar k}{m} \sum_{j=1}^{\infty} \frac{f_{0j}}{\omega_{0j}} \sin(\omega_{0j}t)\,. \tag{4.16}$$

In the latter equality the definition of oscillator strength [Bro+04]

$$f_{0j} = \frac{2m\omega_{0j} |\boldsymbol{r}_{0j}|^2}{3\hbar} \tag{4.17}$$

as well as

$$\sum_{l=x,y,z} (k\hat{\boldsymbol{e}}_l \cdot \boldsymbol{r}_{0j})\, r_{0j,l} = k\left(r_{0j,x}^2 + r_{0j,y}^2 + r_{0j,z}^2\right) = k\,|\boldsymbol{r}_{0j}|^2 \tag{4.18}$$

and again $\boldsymbol{r}_{0j} \in \mathbb{R}$ have been used.

Spectral analysis The Fourier transform [Pre+92]

$$\mathcal{F}[g] = \tilde{g}(\omega) = \int g(t) e^{-i\omega t} dt \tag{4.19}$$

$$\mathcal{F}^{-1}[\tilde{g}] = g(t) = \frac{1}{2\pi} \int \tilde{g}(\omega) e^{i\omega t} d\omega \tag{4.20}$$

of equation (4.16) shows peaks at the excitation energies with amplitudes proportional to the respective oscillator strength. However, due to the finite propagation time T, equation (4.16) is only valid for $0 \le t \le T$. This is described by a window function

$$W_T(t) = \Theta(t) - \Theta(t - T) \tag{4.21}$$

which is multiplied onto the expression of equation (4.16).

The separate Fourier transforms of the sine from equation (4.16) and the window function are

$$\mathcal{F}[\sin(\omega_{0j}t)](\omega) = \frac{i}{2}\left[\delta(\omega - \omega_{0j}) - \delta(\omega + \omega_{0j})\right] \quad \text{and} \tag{4.22}$$

$$\mathcal{F}[W_T](\omega) = \left[\cos\left(\frac{\omega}{2}T\right) + i\sin\left(\frac{\omega}{2}T\right)\right] \frac{\sin\left(\frac{\omega}{2}T\right)}{\frac{\omega}{2}}\,, \tag{4.23}$$

where $\delta(\cdot)$ denotes Dirac's delta distribution.

Due to the convolution theorem $\mathcal{F}[fg] = \mathcal{F}[f] * \mathcal{F}[g]$ the final result for the imaginary part of the Fourier transform of equation (4.16) is

$$-\frac{2m}{3e\hbar k}\Im\left\{\mathrm{Tr}\left[\widetilde{\delta\mu}_\gamma^{(\vartheta)}(\omega)\right]\right\} = \sum_{j=1}^{\infty}\frac{f_{0j}}{\omega_{0j}}\left[\frac{\sin((\omega-\omega_{0j})T)}{\omega-\omega_{0j}} - \frac{\sin((\omega+\omega_{0j})T)}{\omega+\omega_{0j}}\right]. \quad (4.24)$$

It consists of sine cardinal shaped lines at the excitation energies ω_{0j} and $-\omega_{0j} = \omega_{j0}$. The spectrum for $\omega < 0$ is the one for $\omega > 0$ mirrored and with negative strengths. Therefore, it contains no additional information and can be omitted if $\omega_{0j}T \ll 1$ for all excitations $0 \to j$[32]. The latter condition, which is easily fulfilled in practice, ensures that the tails of the sine cardinal shaped lines in the $\omega < 0$ spectrum do not penetrate into the $\omega > 0$ spectrum. Under this assumption, I omit the $\omega < 0$ part in the following and only consider $\omega > 0$ for simplicity.

With the results from equation (4.24) the DSF from equation (4.6) reads[33]

$$S(\hbar\omega) = \sum_{j=1}^{\infty} f_{0j}\frac{\omega}{\omega_{0j}}\frac{\sin((\omega-\omega_{0j})T)}{\pi\hbar(\omega-\omega_{0j})} \overset{T\to\infty}{\longrightarrow} \sum_{j=1}^{\infty} f_{0j}\delta(\hbar\omega-\hbar\omega_{0j})\,. \quad (4.25)$$

The DSF can be calculated from real-time TDDFT according to equation (4.6). Energies and oscillator strengths of excitations can then be fitted according to equation (4.25).

The resulting data of the induced dipole moment are given on an equidistant time grid with a time-step size Δt for $0 \le t \le T = N_t\Delta t$. The FFT algorithm requires the number of time steps to be $N_t^{(\mathrm{ft})} = 2^n$ with $n \in \mathbb{N}$ and $T^{(\mathrm{ft})} = N_t^{(\mathrm{ft})}\Delta t \ge T$. The data from the calculation with $0 \le t \le T$ are usually filled up with zeros up to $T^{(\mathrm{ft})}$ to enforce this constraint.

Due to Nyquist's sampling theorem [Pre+92] the discrete DSF is defined in the interval $-\frac{\pi}{\Delta t} \le \omega \le \frac{\pi}{\Delta t}$. The part of the spectrum with $\omega < 0$ is again the one that I omit. The sampling rate in the frequency domain is determined by $\Delta\omega = \frac{2\pi}{T^{(\mathrm{ft})}}$.

The latter can be used as a simple numerical trick. By attaching zeros to the dipole moment data, one can arbitrarily refine the sampling rate in the frequency domain, which only depends on $T^{(\mathrm{ft})}$. The shape of the spectral lines remains untouched since it only depends on the real propagation time T.

This leads to the refined DSF in figures 4.1 (a3)+(b3) with $\hbar\Delta\omega = 0.0063\,\mathrm{eV}$. In comparison to the original DSF in figures 4.1 (a2)+(b2) with $\hbar\Delta\omega = 0.05\,\mathrm{eV}$, the sampling rate is refined by a factor of eight. This way, the spectra can be analyzed much better and the identification of distinct spectral lines is much easier.

The fit to equation (4.25), which is also shown in figures 4.1 (a3)+(b3), matches the simulated spectrum perfectly for both test systems. The energetic positions of the fitted excitations are indicated by arrows. The evaluated spectra are displayed in figures 4.1 (a4)+(b4) with the single excitations artificially broadened by Gaussian functions.

The Na_4 excitations are named by their symmetry with respect to the coordinate system drawn in figure 4.1 (a1) and their energetic order. Instead of the previously

[32] The spectrum for $\omega < 0$ represents the reverse transitions $j \to 0$.
[33] Note that $\frac{\omega}{\omega_{0j}}\delta(\omega-\omega_{0j}) \to \delta(\omega-\omega_{0j})$.

identified three to five excitations, the refined spectrum reveals 11. The overall spectrum compares well to former TDLDA Casida [CAS95; Cas+98] calculations and the experiment [VÖC99].

The two dominant excitations of BChl$a$ are called Q_y at 1.677 eV with a strength of 0.256 and Q_x at 1.994 eV with a strength of 0.075. Both compare well to the results of [ONS10] who get Q_y at 1.73 eV and Q_x at 2.02 eV for the exact same structure with a density functional tight-binding Hamiltonian. Both are reasonable close to the experiment as published in [CGK06, Figure 3] and [FLC96] with Q_y at 772 nm $\approx$ 1.61 eV and Q_x at 590 nm $\approx$ 2.10 eV. Especially the Q_y excitation plays a major role in natural light harvesting processes, which is discussed in the last section.

Between Q_y and Q_x there are two additional states with a distinct charge-transfer character[34] at 1.739 eV with a strength of 0.018 (CT1) and at 1.863 eV with a strength of 0.012 (CT2). The states CT1 and CT2 are artifacts, which are predicted at too low excitation energies due to the self-interaction error of the TDLDA functional and the missing derivative discontinuity [Toz03; DH04] as discussed in section 2.5.2. They do not appear within this energy range in similar calculations with a tuned RSH functional (see section 5.3.1).

The appearance of spurious excitations in spectra from TDDFT with pure density functionals in BChl$a$ and similar systems is also discussed in the literature [Sun99; Sun00; Sun03; DH04; Cai+06; Qu+09]. For the later discussion about the reliability of TDLDA for the description of energy transfer in aggregates of BChl$a$, these states are of importance. However, one would not have been able to recognize them from the traditional evaluation of real-time dipole spectra in figures 4.1 (b1)+(b2).

At the end of this part, I conclude with the comment that dark excitations with vanishing oscillator strength cannot be seen with the evaluation scheme as presented so far. However, dark states that are dipole-symmetry forbidden can be identified if one chooses an excitation and an observable that do not show dipole symmetry. An example is the excitation that arises from the anti-symmetric coupling between two identical 2-level systems as discussed in section 5.4. This transition is not excited by a weak, global boost and cannot be seen in the total dipole moment. It can still be identified by, e.g., applying the boost in the half-space of one 2-level system and observing the dipole moment in this half-space.

Transition dipole moments The transition dipoles $\boldsymbol{r}_{0j}$ of an excitation are often used in models of coupled chromophores (see section 5.4). Their absolute values are given by the corresponding oscillator strengths through equation (4.17). Their direction is given by the Fourier coefficients of the induced dipole moment vector at the respective transition frequency, i.e.,

$$\boldsymbol{r}_{0j} \propto \widetilde{\boldsymbol{\delta\mu}}(\omega_{0j})\,. \tag{4.26}$$

This can be seen from equation (4.15).

The information about the direction of the transition dipoles can be used to calculate the angles between boost and the transition dipoles, which appear in the factors $\boldsymbol{k}\cdot\boldsymbol{r}_{0j}$ in equation (4.15). This way, one can also calculate the oscillator strengths from a

[34] The transitions could be identified by their transition densities and the analysis of their natural transition orbitals (NTO) [Mar03b] from Q-Chem TDLDA calculations, which are presented in section 5 and appendix E.3.2.

single propagation as long as the initial boost is not orthogonal to any of the transition dipoles. Performing a single calculation can be less accurate for oscillator strengths but is often sufficient, especially if one is only interested in the excitation energies of excitations with reasonable strength.

4.3. Refined transition densities

If the dipole operator in equation (4.12) is replaced by the density operator $\hat{n}$, one gets, after similar considerations, an expression for the imaginary part of the Fourier transformed density fluctuation $\widetilde{\delta n} = \mathcal{F}[n - n^{\mathrm{GS}}]$ (for $\omega > 0$)

$$\Im\left\{\widetilde{\delta n}\left(\boldsymbol{r},\omega\right)\right\} = -\sum_{j=1}^{\infty}\left(\boldsymbol{k}\cdot\boldsymbol{r}_{0j}\right)\rho_{0j}\left(\boldsymbol{r}\right)\frac{\sin\left((\omega-\omega_{0j})T\right)}{(\omega-\omega_{0j})}\,. \tag{4.27}$$

At a certain transition frequency ω_{0n}, this reads

$$\Im\left\{\widetilde{\delta n}\left(\boldsymbol{r},\omega_{0n}\right)\right\} = -T\left(\boldsymbol{k}\cdot\boldsymbol{r}_{0n}\right)\rho_{0n}\left(\boldsymbol{r}\right) - \sum_{\substack{j=1\\ j\neq n}}^{\infty}\left(\boldsymbol{k}\cdot\boldsymbol{r}_{0j}\right)\rho_{0j}\left(\boldsymbol{r}\right)\frac{\sin\left((\omega_{0n}-\omega_{0j})T\right)}{\omega_{0n}-\omega_{0j}} \tag{4.28}$$

with $\lim_{\omega\to\omega_{0j}}\frac{\sin((\omega-\omega_{0j})T)}{\omega-\omega_{0j}} \to T$.

Transition densities of excitations that are energetically close to $\hbar\omega_{0n}$ penetrate into $\Im\left\{\widetilde{\delta n}\left(\boldsymbol{r},\omega_{0n}\right)\right\}$ similar to the spectral lines in figures 4.1 (a3)+(b3). The proportionality in equation (4.8) corresponds to the approximation

$$\Im\left\{\widetilde{\delta n}\left(\boldsymbol{r},\omega_{0n}\right)\right\} \approx -T\left(\boldsymbol{k}\cdot\boldsymbol{r}_{0n}\right)\rho_{0n}\left(\boldsymbol{r}\right)\,. \tag{4.29}$$

Hence, it is only valid for $|\omega_{0n}-\omega_{0j}|T \gg 1$ for all $j \neq n$.

However, the real transition densities can be calculated by inverting equation (4.28) for a closed set of M interfering transitions $0 \to n_{1\ldots M}$. For this finite subset equation (4.28) can be written as a matrix-vector equation with a $M \times M$ correlation matrix $\underline{\boldsymbol{C}}$ and a diagonal normalization matrix $\underline{\boldsymbol{A}}$

$$\mathfrak{S}\left\{\widetilde{\boldsymbol{\delta n}}\right\} = \underline{\boldsymbol{C}}\,\underline{\boldsymbol{A}}\,\boldsymbol{\rho}\,. \tag{4.30}$$

The vector $\mathfrak{S}\left\{\widetilde{\boldsymbol{\delta n}}\right\}$ consists of the $\Im\left\{\widetilde{\delta n}\right\}$ at energies $\hbar\omega_{0n_{1\ldots M}}$ while $\boldsymbol{\rho}$ consists of the pure transition densities $\rho_{0n_{1\ldots M}}$. $\underline{\boldsymbol{A}}$ contains the coefficients $A_{jj} = -T\left(\boldsymbol{k}\cdot\boldsymbol{r}_{0n_j}\right)$. $\underline{\boldsymbol{C}}$ contains the correlation factors $C_{ij} = \frac{\sin\left((\omega_{0n_i}-\omega_{0n_j})T\right)}{(\omega_{0n_i}-\omega_{0n_j})T}$. Therefore, it is symmetric and one on its diagonal. Moreover, $\underline{\boldsymbol{C}}$ only depends on the excitation energies and $\underline{\boldsymbol{A}}$ only on their strengths. The approximation (4.29) corresponds to setting $\underline{\boldsymbol{C}}$ to unity.

The inversion of equation (4.28), which is used for decoupling the transition densities in the following, finally reads

$$\boldsymbol{\rho} = \underline{\boldsymbol{A}}^{-1}\,\underline{\boldsymbol{C}}^{-1}\,\mathfrak{S}\left\{\widetilde{\boldsymbol{\delta n}}\right\}\,. \tag{4.31}$$

$\underline{\boldsymbol{A}}$ is diagonal and is easily inverted while $\underline{\boldsymbol{C}}$ can be inverted by standard algorithms.

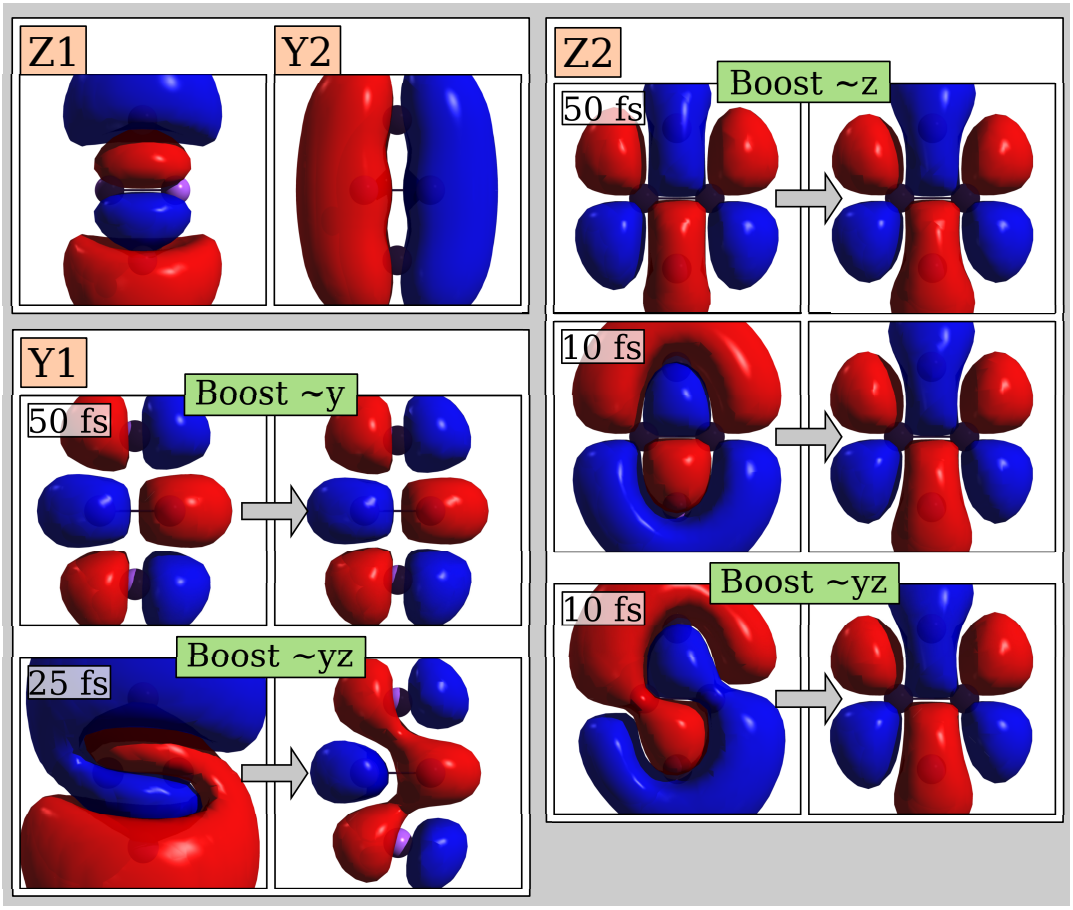


Figure 4.2: Transition densities of the Z1, Z2, Y1, and Y2 transitions of Na_4. The Z2 and Y1 transition densities are displayed for different propagation times and boost directions before and after the correction described in the text. The gray arrows indicate the correction. (iso-surface at $\pm 0.0001\,a_0{}^{-3}$)

The approximated and corrected transition densities are discussed using figure 4.2 by means of the Na_4 cluster and its spectrum in figures 4.1 (a3)+(a4) for different propagation times and boost directions. The latter are chosen parallel to the z-coordinate ($\sim z$), parallel to the y-coordinate ($\sim y$), or along the diagonal in the yz-plane ($\sim yz$). Thus, the excitations with transition dipoles parallel to the x-axis (X1-X3) are always switched off. The Z1 and Y2 excitations, whose transition densities are shown in figure 4.2 (top left block), are the dominant ones and hardly affected by the correction.

According to the spectrum in figure 4.1 (a3) transition densities of the Z1 and, if excited, the Y2 excitations notably penetrate into the Z2 transition density, which is shown in figure 4.2 (right block). The approximated transition densities according to equation (4.29) are shown on the left-hand side inside this block. For a boost parallel to the z-direction and a propagation time of 50 fs the transition density is already well described such that the decoupling (right-hand side inside the block) just adds a small correction. If the propagation time is reduced to 10 fs, the approximated transition density is highly polluted by the Z1 transition density but corrected by the

decoupling. Finally, a boost along the diagonal of the yz-plane additionally excites the Y2 excitation, which breaks the z-symmetry of the approximated Z2 transition density. The correction, however, also removes this effect such that all corrected transition densities match perfectly.

This works the same for the other excitations seen in the spectrum. Still, a difficult benchmark case is the Y1 excitation, which is completely hidden by the Z1 excitation (if it is excited). If the Z1 excitation is switched off by a boost in y-direction, the Y1 transition density is accessible and, after $T = 50\,\mathrm{fs}$, the correction is rather small (figure 4.2, bottom left block). For a boost in yz-direction the Y1 transition density is completely hidden behind the Z1 transition density. The decoupling scheme can remove most of its influence and restore Y1 at least qualitatively.

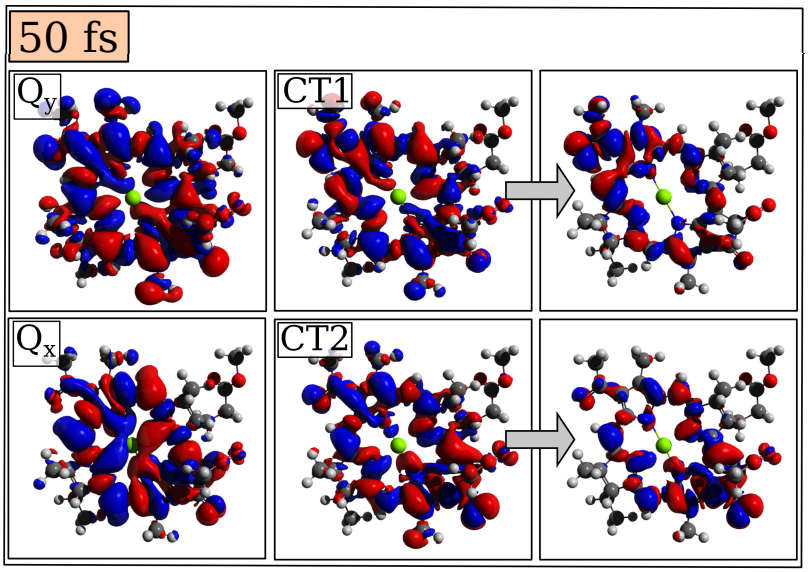


Figure 4.3: Transition densities of the Q_y and Q_x transitions as well as the CT1 and CT2 transitions before and after the correction. The gray arrows indicate the correction. (iso-surface at $\pm 0.0002\,\mathrm{a_0}^{-3}$)

The same scheme works as well for the BChl*a* with the spectrum in figures 4.1 (b3)+(b4). The respective transition densities, which all stem from a single 50 fs propagation, are shown in figure 4.3. The stronger Q_y and Q_x transitions are hardly influenced by the correction with this propagation time. However, the states CT1 an CT2, which are highly polluted especially by the Q_y transition, are decoupled successfully. The corrected transition densities of these two states allow for the comparison with the transition densities from Q-Chem TDLDA calculations and their identification. They show increased intensity in the top left (CT1) and bottom right (CT2) corners inside the figure.

Finally, to measure the quality of the corrected transition densities, one can check how well the transition density reproduces the transition dipole from the spectral evaluation, i.e.,

$$\boldsymbol{r}_{0j} \stackrel{!}{=} \int \boldsymbol{r}\rho_{0j}(\boldsymbol{r})d^3r\,. \tag{4.32}$$

If the left-hand side and the right-hand side of equation (4.32) result in two vectors

with different directions, the correction did not work well as for the Y1 excitation of Na_4[35]. If the left-hand side and the right-hand side result in two vectors with almost identical directions but amplitudes that differ by some factor C, probably only the scaling with the matrix $\underline{\boldsymbol{A}}^{-1}$ did not work well since the oscillator strength used within $\underline{\boldsymbol{A}}$ was not accurate enough.

The latter can be used to refine the strength and therefore the amplitude of the transition density of this excitation. If, e.g., $\boldsymbol{r}_{0j} = C \int \rho_{0j} \boldsymbol{r} d^3r$, the improved transition dipole is $\frac{1}{\sqrt{C}} \boldsymbol{r}_{0j}$, which leads, after renormalization with the improved $\underline{\boldsymbol{A}}^{-1}$, to a consistently improved transition density $\sqrt{C}\rho_{0j}$.

As final remark in this section, I want to emphasize that many of the excitations of the presented test systems cannot be identified with the traditional evaluation of real-time spectra and typical propagation times. The techniques presented in this section allow for a quantitative evaluation with even shorter propagation times.

[35] For the Y1 excitation of Na_4 the right-hand side of equation (4.32) from the transition density in figure 4.2 (Y1, boost $\sim yz$, corrected) still has a significant contribution in z-direction whereas the left-hand side of equation (4.32) from the spectral evaluation only has a y-component, as expected.

5. Excitation dynamics between bacteriochlorophylls

In view of the increasing energy demands of humanity and the demand for renewable energy sources such as solar energy, the investigation of natural light-harvesting systems remains in the focus of science. Nature has developed a highly efficient machinery for light absorption, excitation-energy transfer (EET), and charge separation while protecting the participating chromophore-protein structures from being damaged by the incoming radiation. All of these properties are also relevant for artificial light harvesting. Among the different photosynthetic organisms, i.e., plants, algae, and different kinds of bacteria, photosynthetic purple bacteria belong to the ones best studied. Recent reviews on this topic are given in [Hu+98; Hu+02; CGK06; SR06; CF09; CK10; Fle+12; CM16].

The photosynthetic apparatus of purple bacteria consists of highly structured chromophore-protein complexes that are arranged in the cytoplasmic membrane of these bacteria. Sun light is absorbed by ring-shaped antenna complexes called LH2. These transfer the related excitation energy towards reaction centers (RC), which are embedded inside of larger, ring-shaped complexes called LH1. The RC uses the excitation energy to ultimately generate a proton gradient between the two sides of the membrane. This gradient is the driving force for the synthesis of ATP, which acts as energy source inside the organism. [CGK06; Hu+02]

To understand the complex processes within the photosynthetic apparatus of purple bacteria requires the collaboration between different fields from physics, chemistry, and biology as well as a combination of experimental studies and theoretical models. Especially in view of the highly efficient EET inside the LH2 complex of *Rbl. acidophilus*, which is the subject of this section, a detailed theoretical analysis is mandatory to reveal the underlying mechanisms. This includes the electronic coupling mechanisms between the chromophores inside the complex and their interaction with the surrounding environment. [Hu+02; CGK06; Eng+07; CF09; Fle+12; SŞS12]

Modern TDDFT is well suited for the description of large molecules or molecular aggregates due to its good ratio between efficiency and accuracy. The simulation of the whole LH2 complex or its main chromophores with thousands of valence electrons is not possible so far. However, it comes into reach due to the increasing computational power of modern computers and parallel codes such as BTDFT. [Jor+15; CM16]

Still, the simulation of the LH2 (or its chromophores) as one quantum system by means of TDDFT is only possible with a light-weight xc functional such as TDLDA. These have well known deficiencies such as their inability to describe excitations with a charge-transfer character (see section 2.5) [Toz03; DH04]. I already mentioned this in the last section when I showed the spectrum of a BChl$a$ chromophore with two spurious CT[36] excitations (CT1 and CT2) in figures 4.1 (b3)+(b4).

In view of the simulation of EET within the LH2 complex from real-time TDDFT, the first question is therefore how well light-weight functionals such as TDLDA are suited for its description. This is directly connected to the question of how to describe EET using real-time TDDFT since its only reliable output is the time-dependent density $n(\boldsymbol{r}, t)$. The chromophores are embedded in a protein scaffold, which leads to their distortion and influences their coupling to each other [CGK06; Pap+03; AHD16].

[36] I label those excitations with 'CT'. However, they only have some charge-transfer character.

Thus, a further question is how much of the environment must be included explicitly into the TDDFT calculation and which part can be modeled.

I introduce the LH2 complex of *Rbl. acidophilus* in section 5.1 and the modelling of the environment in proximity to the relevant chromophores by an electrostatic potential in section 5.2. The following sections address the first steps towards a first-principles simulation of EET within the LH2 complex with real-time TDDFT and, in particular, the questions mentioned above. Therefore, I investigate the description of the relevant excitations of a single BChl*a* and two aggregated BChls within the so-called B850 ring inside the LH2 complex in section 5.3. This includes a discussion about the influence of the environment, which is treated by an electrostatic potential or partly directly within the TDDFT calculation.

Section 5.4 is dedicated to the general description of real-time energy transfer and the prediction of coupling strengths from real-time TDDFT. In section 5.5, the latter is applied to a donor-acceptor system consisting of two resonant sodium dimers or two BChl*a* molecules, respectively. These sections report the current state of my work, which is concluded in section 5.6.

5.1. The LH2 complex of Rhodoblastus acidophilus

The LH2 complex of purple bacteria, here in particular of *Rbl. acidophilus* (Strain 10050), is subject to intense investigations since many years. Its study experienced a boost after revealing detailed structural information from X-ray analysis [Küh95; McD+95]. These structural data undisclosed the arrangement of the embedded chromophores, i.e., BChls and carotenoids, which play a dominant role in light-harvesting processes. These chromophores are anchored in an apoprotein scaffold. The structure of [McD+95] has been refined by [Pap+03] who revealed new structural information. The structure used in this thesis by [GC] shows the same structural features as the one from [Pap+03] but with a better resolution of 1.85 Å. [Fre+96; Hu+02; CGK06; CF09]

The LH2 complex of *Rbl. acidophilus* has a ring-shaped structure with a nine-fold symmetry. The main EET is mediated through BChl*a* molecules, which are organized in two rings called B800 and B850, corresponding to the wavelength of their maximum in-vivo absorption (in nm). The latter are displayed in figure 5.1 (a) without the apoprotein scaffold and the carotenoids. The nine B800 BChls are separated from each other while the 18 B850 BChls build a close aggregate. This aggregation leads to a strong coupling between the B850 BChls, which lowers the excitation energy of the lowest dipole-active state and shifts the absorption towards red. The B850 ring is the one discussed in this work. [Hu+02; CGK06]

The apoprotein scaffold builds rod-like structures inside (α) and outside (β) the BChl rings. These rods are nearly perpendicular to the drawing plane of figure 5.1 (a) (not shown). Every second B850 BChl is ligated by an α-apoprotein (α-BChl), the other half by a β-apoprotein (β-BChl). A pair of one α-BChl and one β-BChl builds a dimer. α-BChls and β-BChls are chemically identical but are distorted differently due to their different local environments. [Pap+03; CGK06]

In figure 5.1 (a), the BChls with residue labels BCL-301, BCL-302, and BCL-303 (in the following only B301, B302, and B303) are pointed to by arrows. B301 and B303 are α-BChls whereas B302 is a β-BChl. In figure 5.1 (b), these BChls are

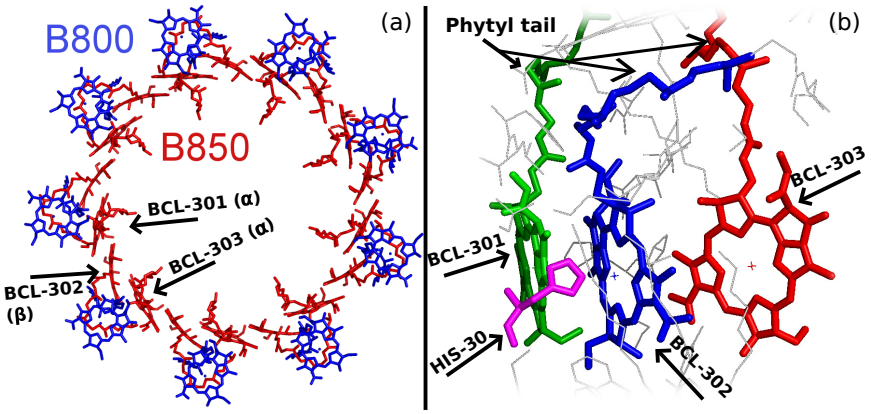


Figure 5.1: (a) BChl*a* molecules inside the LH2 complex of *Rbl. acidophilus* organized in two rings called B800 (blue) and B850 (red). (b) Closeup of the BChl*a* molecules BCL-301, BCL-302, and BCL-303 together with the HIS-30 ligand, which coordinates the central magnesium atom of BCL-302. (The figures were generated with PyMOL [Sch15].)

shown together with some of their surrounding environment (white) and especially the histidine residue HIS-30 (purple). The latter is part of a β-apoprotein and coordinates the central magnesium atom of B302. Also indicated in figure 5.1 (b) are the phytyl tails [CGK06; Fre+96] of the BChls, which play a mainly structural role [Fre+96] and are truncated for the TDDFT calculations. This is described in appendix F.2 and commonly done in the literature, e.g., [Sun00; VB07; ONS10; OS11; KN13].

Figure 5.2 shows a single BChl*a* (B302) once from the front (a) with the Q_y and Q_x transition dipole moments indicated as arrows according to [CGK06] and once along the Q_y transition dipole (b) with the additional histidine (HIS-30) ligand as used for the calculations presented below. The numbering of the pyrrol rings within the bacteriochlorin matches the IUPAC standard [Mos87; CGK06][37]. The position at which the phytyl tail has been replaced by a hydrogen atom is marked in both figures.

The Q_y and Q_x transition dipoles are almost perpendicular to each other. The BChls in the B850 ring are oriented such that their Q_y transition dipoles lie roughly inside the drawing plane, their Q_x dipoles are perpendicular to the plane. Two neighboring B850 BChls are oriented in a head-to-head or tail-to-tail alignment. [CGK06; Hu+02]

The designation of the Q-band excitations Q_y and Q_x goes back to Gouterman's four-orbital model [Gou61], which explained the strong absorption bands of BChl*a* in the visible spectrum and near infrared (Q-band) and at higher energies (Soret or B-band). The strong Q_y dipole approximately connects the pyrrol rings (I) and (III) as indicated in figure 5.2 (a) and shows its absorption at about 772 nm $\approx$ 1.61 eV in an organic solvent [CGK06, Figure 3]. In the protein environment of the LH2 complex and with other BChls in close proximity, the absorption shifts to towards $\approx$ 800 nm (B800) and $\approx$ 850 nm (B850) [CGK06, §3.2].

The Q_x dipole of BChl*a* is much weaker and shows absorption at about 590 nm $\approx$

[37] Note that the BChl*a* from section 4 in figures 4.1 (b1) and 4.3 are viewed from the opposite side. There, the pyrrol ring (I) is top left and the pyrrol rings (III) and (V) are bottom right and the numbering is counter clockwise.

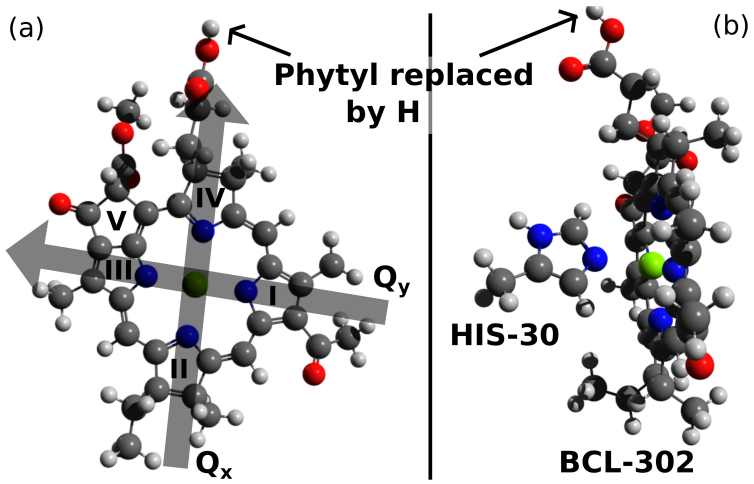


Figure 5.2: (a) B302 front view with Q_y and Q_x transition dipoles according to [CGK06] and (b) view along the Q_y dipole with the coordinating histidine. Carbon is gray, hydrogen white, oxygen red, nitrogen blue, and magnesium green. The Q_y dipole connects the pyrrol rings (I) and (III) while Q_x connects (II) and (IV). (The figures have been generated with Avogadro [Han+12].)

2.10 eV in the same solvent [CGK06, Figure 3]. The B-band is not relevant for the EET in the B850 ring [Hu+02, §3.1.1] and is not discussed here as well as the carotenoids, which build a second class of chromophores inside the LH2.

The different energy pathways inside the LH2 complex are explained in much detail together with the corresponding experimental and theoretical references in [CGK06, §6] and [Hu+02, §3]. At some point after the absorption of sun light, the related excitation energy is transferred onto the B850 ring or directly absorbed by B850 BChls. The excitation energy then travels around the B850 ring and can be transferred onto neighboring LH2 complexes or the LH1 towards the RC.

Experimental evidence was found [Eng+07; SŞS12] that the EET around the B850 ring gains its efficiency from exploiting quantum mechanical coherence, i.e., excited states that are delocalized among a couple of B850 BChls [Dah+01; CGK06; SR06]. Within an exciton Hamiltonian picture [CGK06, §5], the delocalization length of an excitation depends on a number of factors such as static and dynamic disorder. This is of special interest since light harvesting takes place close to room temperature.

The ratio between the coupling strength between the BChls and the energetic disorder determines if the EET can be described as an incoherent hopping process or a coherent wave-like transport. Different theoretical descriptions, which are summarized in [CGK06, Table 4], have been used to estimate coupling strengths and disorder and lead to partly different results [CGK06]. More recent model Hamiltonian approaches are reported in [LK06; Veg+15].

The underlying EET mechanisms in the B850 ring of the LH2 complex remain a matter of strong interest [CGK06; CF09; Fle+12; SŞS12]. A description with real-time TDDFT, which treats the whole (or a part of) the LH2 complex intrinsically as one quantum system and is flexible enough to include environment effects, can help

to get further insights into this process. This section only sets the first steps towards a TDDFT simulation of the EET processes in the LH2 complex and tries to make some comments on the feasibility of this approach. Another approach to reveal the real-time excitation dynamics in LH2 is reported in [Veg+15].

5.2. Modelling the environment

5.2.1. Electrostatic environment potential and ligands

The B850 ring resides in the protein scaffold of the LH2 complex with other chromophores in its proximity. The LH2 complex, again, is embedded in a lipid-bilayer membrane. The environment of a BChl*a* tunes its absorption by the coupling to other chromophores, coordination by ligands such as histidine, or screening effects. On the other hand, vibrational states interact with the B850 BChls and influence the EET. [CGK06; Fle+12; Veg+15]

In this thesis, the environment is described as a dielectric medium with static, partial charges sitting on the atoms in the proximity to the investigated chromophores and free charges in the surrounding solvent [UB14]. This is processed into an environment potential, which can be included as an external potential into the TDDFT calculation and modifies the ionic potential from the simulated molecules. The environment potential, which is unique for each of the different molecular setups discussed below, was generated by Johannes Förster from the group of Prof. Matthias Ullmann (Computational biochemistry, University of Bayreuth). The details of the environment potential and the structure preparation are explained in appendix F.

I want to make clear that the environment potential is the only quantity that enters the TDDFT calculations termed 'with environment' below. The dielectric medium is not included into the solution of Poisson's equation for the Hartree potential. The static polarization of the surrounding medium due to the presence of the B850 BChl (or BChls) is represented by the reaction field [UB14] inside the environment potential.

Since the environment as described by the potential is static but dynamic effects in the electronic coupling to neighboring ligands can be important, I additionally included the respective coordinating histidine residues directly into some of the simulations. For B302, this leads to the setup as shown in figure 5.2 (b). To this end, I truncated each histidine residue from its protein backbone, which is also described in appendix F.

5.2.2. Influence on the electronic ground state

The influence of the environment potential on the ground-state properties of a single B850 BChl (B302) was investigated by Nikolaj Swiridoff [Swi15]. I summarize the most important results in the following.

The environment potential had only negligible effects in this study. Nikolaj found that the KS eigenvalues are hardly affected by the potential. The changes with environment potential are close to those one would get from varying numerical parameters such as the grid spacing. Moreover, the energetic order of the KS orbitals around the HOMO is unchanged. Since a KS eigenvalue determines the asymptotic exponential decay of its corresponding KS orbital [Kre+98], the electronic structure of the B850 BChl as seen from outside is not influenced strongly by the environment potential.

In total, this means that the environment potential as used here does not lead to dramatic changes but is considered as a small correction. This is also expected since the environment tunes the absorption of the chromophores but does not change it fundamentally [CGK06; Jor+15]. The influence on the ground-state quantities is small. Yet, the potential has an important effect on the excitation structure of the B850 BChls from real-time TDLDA, which is discussed in the following.

5.3. Spectra of B850 bacteriochlorophylls from TDLDA and ωPBE

5.3.1. Spectra of single bacteriochlorophylls

In the past, the reliability of the Gouterman model for porphyrins [Gou61] was questioned [Sun99; Sun00; CSR02] due to the appearance of additional, weak states from TDDFT calculations in the energy range of the Q-band and between Q- and B-band. These states were found to have a charge-transfer character as the CT1 and CT2 states from the real-time TDLDA spectrum of the BChl*a* in section 4.2. They are usually found at energies above the B-band as predicted by other methods [Cai+06; LK06].

The energetic downshift of states with charge-transfer character was explained by [Toz03] and [DH04] and attributed to the missing derivative discontinuity and the self-interaction error of pure density functionals. This wrong description of excitations was shown explicitly for porphyrins [CSR02] and chlorophylls [DR05]. However, functionals have been developed that describe charge transfer in a much better way [CM16]. These are the CAM-B3LYP functional [Cai+06; YTH04] and tuned RSH functionals [SKB09b; SKB09a; BLS10; Kar+11; Kör+11; Kro+12; KKK13][38].

Therefore, the TDLDA spectra below are compared to Q-Chem [Sha+15] calculations with the tuned ωPBE functional, which is of the latter type and expected to deliver reliable results [SKB09b; SKB09a; TEE10; Moo+15]. The range separation parameter ω was tuned as described in appendix E.2.3. All Q-Chem calculations were performed by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil).

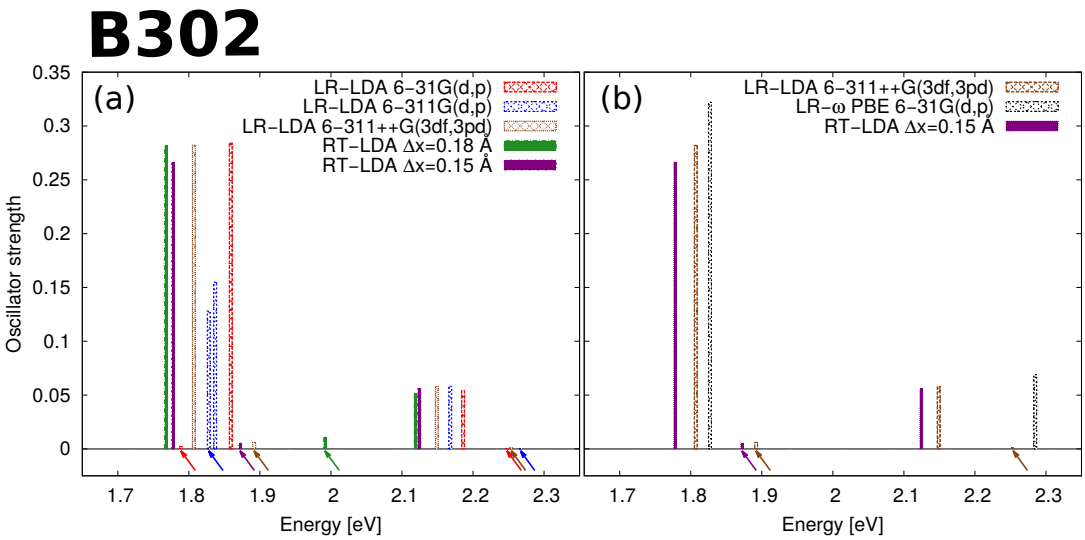
Transition densities and natural transition orbitals (NTO) [Mar03b] of the excitations discussed below are attached to this work in appendix E.3.2 as supporting information. These were used to identify excitations and determine their charge-transfer character[39].

The stereotypes of excitations are those already identified for the BChl*a* from section 4 with transition densities in figure 4.3 on page 53. What I termed CT1 excitation in that section has a strong transition density on the pyrrol ring (I) while the CT2 excitation has a strong transition density on the pyrrol rings (III) and (V).

The spectrum of B302 is shown in figure 5.3 from real-time (RT, solid) and Q-Chem linear-response (LR, dashed) calculations with the TDLDA functional for different grid spacings or basis sets, respectively. The evaluation of the real-time spectra followed section 4.2. The excitation energies and strengths were obtained with the same quality as those from section 4.2 (see figure 4.1). The excitations are labeled by S_n in their energetic order.

[38] Also see [Moo+15] for a discussion of charge transfer and charge-transfer excitations in TDDFT.

[39] NTOs describe an excitation as an electron-hole transition. A charge-transfer-like excitation shows electron and hole orbitals that are spatially separated while those of a non-charge-transfer excitations occupy the same space.



Data

	RT-TDLDA		LR-TDLDA			LR-ωPBE
Excit. state	$\Delta x = 0.18$ Å	$\Delta x = 0.15$ Å	6-31G(d,p)	6-311G(d,p)	6-311++ G(3df,3pd)	6-31G(d,p)
S_1	1.768 (0.282)	1.778 (0.266)	1.789 (0.002)	1.828 (0.128)	1.807 (0.282)	1.827 (0.322)
S_2	1.992 (0.011)	1.872 (0.005)	1.859 (0.284)	1.837 (0.155)	1.892 (0.006)	2.285 (0.069)
S_3	2.119 (0.052)	2.125 (0.056)	2.186 (0.054)	2.168 (0.058)	2.149 (0.058)	–
S_4	–	–	2.248 (0.000)	2.267 (0.000)	2.254 (0.001)	–

Figure 5.3: Real-time (RT, solid) TDLDA and Q-Chem linear-response (LR, dashed) TDLDA and ωPBE singlet spectra for B302 with different grid spacings or basis sets. (a) TDLDA results with different parameters, (b) best TDLDA results compared to LR-ωPBE. Energies are in eV, strengths in parenthesis. Q-Chem calculations were done by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil).

The real-time spectrum of B302 with $\Delta x = 0.18$ Å shows the Q_y excitation at 1.768 eV, the Q_x excitation at 2.119 eV, and one additional, weak excitation at 1.992 eV. The latter can be identified with the CT2 excitation (see figure E.3).

If the grid spacing is refined to $\Delta x = 0.15$ Å, Q_y and Q_x excitations are quite stable but slightly blue-shifted. The CT2 excitation, on the other hand, reacts more sensitive. It is red-shifted by ≈ 0.12 eV and shows about half the strength. Further refinement of the grid spacing or other parameters does not lead to major changes, which is shown in appendix E.3.1. I already showed in section 4.2 that the energies of the Q-band excitations of BChl*a* are reasonably well described by TDLDA as compared to the experiment. I expect this to hold true also for the single B850 BChls with a slightly different structure.

The Q-Chem TDLDA spectra were calculated using Gaussian basis sets as implemented in Q-Chem [Sha+15]. The basis sets used for the smaller atoms were 6-31G(d,p) (small, standard), 6-311G(d,p) (medium), and 6-311++G(3df,3pd) (very large). For magnesium, the EPC-LANL2DZ basis set was used.

The calculation with the 6-31G(d,p) basis set predicts the Q_y excitation at 1.859 eV and Q_x at 2.186 eV. The CT2 excitation is predicted below Q_y at 1.789 eV. CT1 can be identified at 2.248 eV with overall tiny strength (see figure E.4). If the CT1

transition is present in the real-time calculations, it can probably not be identified due to its small strength.

If a larger basis set is used, the CT1 and Q_x excitations remain stable up to a small red-shift of 0.02 eV of 0.04 eV, respectively. The Q_y and CT2 excitations, on the other hand, mix strangely if one uses the 6-311G(d,p) basis set. With the 6-311++G(3df,3pd) basis set, they are again separated but with Q_y below CT2 as predicted by the real-time calculation. For illustration, the NTOs for the states S_1 and S_2 are shown in figures E.7, E.8, and E.9 for the three basis sets.

The final Q_y energy is predicted at 1.807 eV and Q_x at 2.168 eV, which is systematically blue-shifted by < 0.03 eV with respect to the real-time calculation with $\Delta x = 0.15$ Å. This also holds true for the CT2 state. The remaining shift can be attributed to further basis-set issues or the principle difference between the Q-Chem and BTDFT ways of treating core electrons, i.e., explicitly or through pseudo potentials.

The CT2 state is highly sensitive to variations of the grid spacing or the basis set. The 6-311G(d,p) basis set even predicts a mixing between CT2 and Q_y, which results in two mixed states with similar oscillator strengths.

The best results of both TDLDA approaches are compared to a calculation with the tuned ωPBE functional and 6-31G(d,p) basis set in figure 5.3 (b). The ωPBE calculation shows no spurious states but only the expected Q-band transitions (see figures E.5 and E.6). Q_x is blue-shifted with respect to all TDLDA calculations. Q_y is shifted by 0.049 eV with respect to the best real-time TDLDA calculation. Since the ωPBE calculation is computationally expensive, using a larger basis set is difficult. However, if one assumes the same red-shift as for the Q-Chem TDLDA when the basis set is improved, the gap between the Q_y excitations as predicted by real-time TDLDA and Q-Chem ωPBE closes completely.

In total, the important Q_y and Q_x states are well described by real-time and Q-Chem TDLDA. However, in Q-Chem, a really large basis set is necessary to converge the Q_y energy and describe the transition qualitatively correct. The oscillator strengths from TDLDA and ωPBE are reasonably close to each other.

The latter is also true for the transition dipoles. The angle between Q_y and Q_x transition dipoles as predicted by Q-Chem ωPBE is 76.7°, which matches perfectly with the 76.6° from the real-time calculation at $\Delta x = 0.18$ Å[40].

The meaning of the spurious states CT1 and CT2 in view of the EET is a matter that requires more investigation. This is especially true since the the pyrrol rings (I) as well as (III) and (V), which show high CT1 and CT2 transition densities, are the ones that overlap between neighboring B850 BChls (see figures 5.1 and 5.2). Therefore, CT1 and CT2 could contribute substantially to the coupling between the BChls and influence the EET simulation, even if they show weak dipole strengths.

Finally, the spectra of B303 as well as real-time spectra of B301 are shown in appendix E.3.1. B303 transition densities from real-time TDLDA and Q-Chem ωPBE are displayed in figures E.11 and E.12. The details differ a little, e.g., that B301 and B303 both show the CT1 and CT2 states with non-vanishing strengths. Yet, the overall message remains the same.

[40] I estimated the error of the real-time evaluation scheme in this case to $\approx 0.4°$ by comparing results from two calculations with different boost directions.

5.3.2. Spectra of two aggregated bacteriochlorophylls

The excitation pattern of aggregated BChls is often described by means of an exciton Hamiltonian approach in which the Q_y transitions of the single chromophores are coupled with a certain coupling strength [CGK06; SR06]. This way, one expects that the spectrum of two aggregated BChls shows a line splitting according to Davydov [Dav64]. I go more into the details of a similar model in section 5.4.1.

The Q_y dipoles of neighboring B850 BChl are almost parallel to each other and build a kind of J-aggregate [KRA65; SR06]. Similar to a coupled pendulum, one expects one transition that corresponds to a symmetric coupling between the Q_y transitions of the single chromophores with almost twice the dipole strength and one transition that corresponds to an antisymmetric coupling with almost vanishing strength. In the J-aggregate, the symmetric transition has lower energy such that the aggregate's absorption is red-shifted with respect to the single chromophores.

The spectrum of the combined B302-B303 system is shown in figure 5.4 from real-time TDLDA, Q-Chem TDLDA, and Q-Chem ωPBE. The Q-Chem calculations for the combined system were only possible with the 6-31G(d,p) basis set.

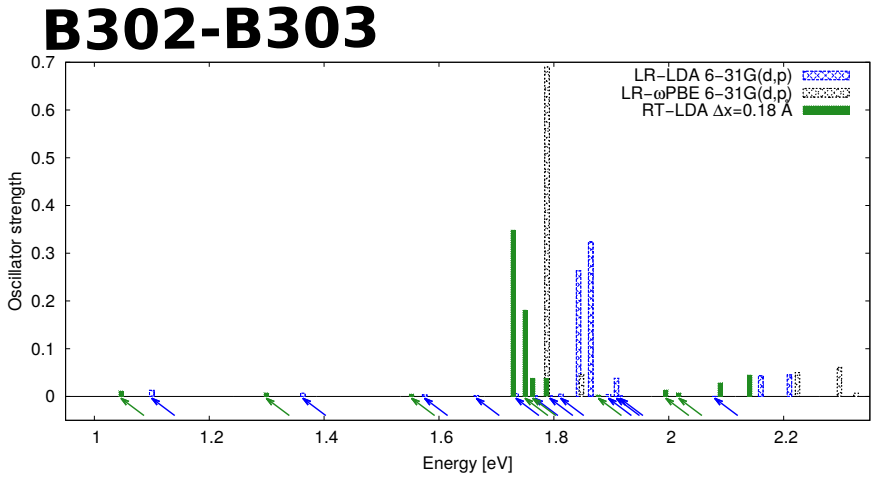
The ωPBE calculation shows the expected behaviour as described above. The Q_y transitions split up into two transitions. The one at lower energy 1.788 eV shows a much larger strength than the one at higher energy 1.848 eV. The NTOs of both states are displayed in figure E.17 on page 146. The electron and hole orbitals occupy the same space as expected from an excitation without charge-transfer character.

The real-time TDLDA and Q-Chem TDLDA calculations predict many artificial states and the deviation between ωPBE and TDLDA results becomes more pronounced. Some of the additional, spurious states are far below the Q_y energy, others are mixed with the Q_y states of the single chromophores. In the real-time calculation, this results in four excitations S_4-S_7 that are within 0.06 eV very close in energy. The transition densities of the two largest real-time transitions S_4 and S_5 are shown in figure E.13 on page 143. S_4 is best represented by the two symmetrically coupled Q_y transitions. S_5 seems to be a mixed state between B302-Q_y and B303-CT1.

The NTOs of the states S_1-S_{10} from Q-Chem TDLDA are shown in figures E.14, E.15, and E.16 on pages 144 to 146. The weak transitions S_1-S_4 at low energies show electron and hole orbitals on different BChls and hence a strong charge-transfer character of the excitation. The transitions S_1-S_3 from Q-Chem TDLDA and real-time TDLDA have similar energies with the systematic shifts already seen from the single B302. Therefore, they can be identified with each other. The electron and hole orbitals from the Q-Chem TDLDA transitions S_5-S_{10} also show a charge-transfer character and often a high amplitude on the overlapping pyrrol rings. This again indicates the participation of the former CT1 and CT2 states in the transitions of the coupled system.

The two excitations with the respective highest strengths from real-time TDLDA at $\Delta x = 0.18$ Å and Q-Chem TDLDA at 6-31G(d,p) differ again by about 0.1 eV. If one assumes the same spectral shifts of the real-time and Q-Chem data as for B302 when improving the grid spacing and the basis set, one again expects real-time TDLDA and Q-Chem TDLDA data to converge.

In conclusion, the coupled B302-B303 system shows spurious, weak excitations at low energies with a strong charge-transfer character as well as excitations that are the



Data

	RT-TDLDA	LR-TDLDA	LR-ωPBE
Excit. state	$\Delta x = 0.18$ Å	6-31G(d,p)	6-31G(d,p)
S_1	1.047 (0.012)	1.100 (0.013)	1.788 (0.690)
S_2	1.299 (0.008)	1.363 (0.007)	1.848 (0.046)
S_3	1.552 (0.005)	1.575 (0.003)	2.224 (0.050)
S_4	1.729 (0.348)	1.665 (0.002)	2.297 (0.061)
S_5	1.750 (0.181)	1.734 (0.006)	2.326 (0.007)
S_6	1.763 (0.038)	1.767 (0.001)	–
S_7	1.786 (0.037)	1.793 (0.001)	–
S_8	1.878 (0.004)	1.812 (0.005)	–
S_9	1.995 (0.014)	1.843 (0.263)	–
S_{10}	2.018 (0.008)	1.864 (0.324)	–
S_{11}	2.089 (0.029)	1.895 (0.004)	–
S_{12}	2.141 (0.045)	1.909 (0.038)	–
S_{13}	–	1.915 (0.001)	–
S_{14}	–	2.080 (0.000)	–
S_{15}	–	2.160 (0.043)	–
S_{16}	–	2.210 (0.045)	–

Figure 5.4: Real-time (RT, solid) TDLDA and Q-Chem linear-response (LR, dashed) TDLDA and ωPBE singlet spectra for the combined B302-B303 system with different grid spacings or basis sets. Excitation energies are in eV, oscillator strength in parenthesis. Q-Chem (LR) calculations were done by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil).

Q_y states mixed with charge-transfer-like excitations. In view of EET simulations, the former are probably less important since one can simulate an excitation that is tuned to only excite a narrow energy band around the Q_y states. Still, the appearance of mixed states makes the applicability of TDLDA for qualitative EET simulations questionable.

5.3.3. Influence of the environment on the spectra

Finally, I investigate the influence of the environment on the spectra of the single B302 and the coupled B302-B303 system. The environment is treated by the environment potential as introduced in section 5.2.1. The related spectra are shown in figure 5.5. The data sets without environment (green data) are the ones that I already discussed in the previous sections. In the spectra with environment, the environment is once treated entirely by the environment potential (red data) and once histidine residues are part of the TDDFT simulation and excluded from the potential (blue data).

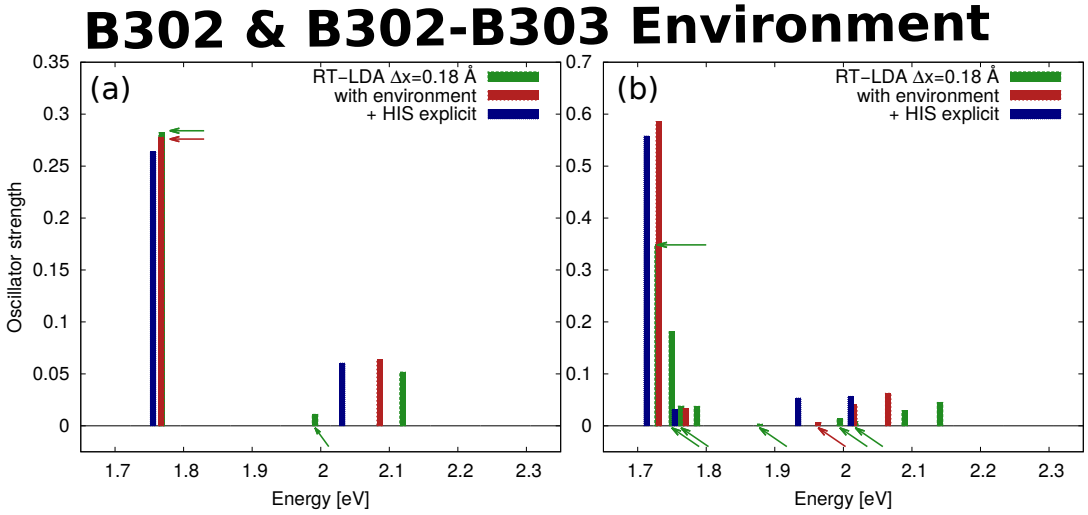
The overall effect of the environment potential on the Q-band states of the single B302 is rather small. The Q_y transition is hardly influenced by the pure environment potential. Treating the histidine directly within the TDDFT calculation shifts its energy by $\approx 0.01\,\mathrm{eV}$ towards red. The Q_x transition reacts a little stronger and is red-shifted by $\approx 0.03\,\mathrm{eV}$ and $\approx 0.09\,\mathrm{eV}$, respectively. However, the critical CT2 state, which is predicted between Q_y and Q_x without environment, vanishes if the environment is included.

The same effect is encountered in the coupled B302-B303 system. The states S_1 and S_2, which were assigned to the charge-transfer excitations between B302 and B303, remain at the lower energy end (not shown in the graphic). However, the calculations with environment potential show the correct excitation pattern for the coupled Q-band states. Just the calculation in which the environment is entirely treated by the potential (red data) still shows one small excitation in the Q-band. The latter disappears if the histidine residues are treated directly within the TDDFT calculation.

This result is important for two reasons: First, including the electrostatic environment and the histidine into the TDDFT calculation seems to remove the most critical spurious excitations in the Q-band systematically. Second, since the Q_y excitation of a single BChl*a* is hardly affected by the environment in the way it is treated here, one can now investigate the role of the spurious states for the EET as simulated with TDLDA. The latter can be done by performing EET simulations with and without environment and comparing the results. If there is no difference, the spurious states are probably less important. If both simulations differ significantly, including the environment potential seems to resolve the issues that are related to TDLDA in this case.

5.4. Coupling strengths and real-time energy transfer

In a real-time simulation of EET one would like to excite a system that consists of several chromophores and visualize and quantify the energy pathway [Hof+13]. However, at this moment the question arises about a physically meaningful energy density or an energy per chromophore.



Data

	RT-TDLDA, $\Delta x = 0.18$Å		
Excit. state	No Env.	Env. pot.	Env. pot. HIS via TDDFT
(a)	**B302**		
S_1	1.768 (0.282)	1.768 (0.278)	1.756 (0.264)
S_2	1.992 (0.011)	2.086 (0.064)	2.033 (0.060)
S_3	2.119 (0.052)	–	–
(b)	**B302-B303**		
S_1	1.047 (0.012)	0.978 (0.013)	1.037 (0.013)
S_2	1.299 (0.008)	1.352 (0.010)	1.274 (0.009)
S_3	1.552 (0.005)	1.731 (0.585)	1.462 (0.003)
S_4	1.729 (0.348)	1.770 (0.033)	1.713 (0.558)
S_5	1.750 (0.181)	1.963 (0.006)	1.754 (0.031)
S_6	1.764 (0.038)	2.016 (0.040)	1.934 (0.054)
S_7	1.787 (0.037)	2.065 (0.063)	2.010 (0.057)
S_8	1.878 (0.004)	–	–
S_9	1.995 (0.014)	–	–
S_{10}	2.018 (0.008)	–	–
S_{11}	2.089 (0.029)	–	–
S_{12}	2.141 (0.045)	–	–

Figure 5.5: Real-time (RT) TDLDA singlet spectra for B302 and the combined B302-B303 system without environment (No Env.), with environment potential (Env.), and with environment potential but histidine ligands included in the TDDFT calculation (Env. & HIS) for $\Delta x = 0.18\,$Å. Excitation energies are in eV, oscillator strength in parenthesis.

Indeed, there are several natural candidates: The first one is an energy density that can be defined as

$$e(\boldsymbol{r},t) = \sum_{\sigma=\uparrow,\downarrow}\sum_{j=1}^{N_\sigma} \varphi_{j\sigma}^*(\boldsymbol{r},t)\hat{H}_{\mathrm{KS}}\varphi_{j\sigma}(\boldsymbol{r},t) - \left[\frac{1}{2}v_{\mathrm{H}}(\boldsymbol{r},t) + v_{\mathrm{xc}}(\boldsymbol{r},t)\right] n(\boldsymbol{r},t) + e_{\mathrm{xc}}(\boldsymbol{r},t)\,. \tag{5.1}$$

This expression is, in an adiabatic way, evaluated with the time-dependent density and the time-dependent KS orbitals. $e_{\mathrm{xc}}[n]$ is the adiabatic xc energy density. Expression (5.1) integrates to the total energy [Cap06] as defined in an adiabatic way with the ground-state energy functional evaluated with the time-dependent quantities. The total excitation energy per chromophore can be defined by

$$\delta E_i(t) = \int_{V_i} e(\boldsymbol{r},t) - e^{(\mathrm{GS})}(\boldsymbol{r}) d^3r\,, \tag{5.2}$$

where V_i is the space associated with chromophore i and $e^{(\mathrm{GS})}$ is the ground-state energy density according to equation (5.1).

Another, rather pragmatic measure is the time-dependent induced dipole moment of the single chromophores, i.e., $\boldsymbol{\delta\mu}_i(t) = \int_{V_i} \boldsymbol{r}\delta n(\boldsymbol{r},t) d^3r$ [Hof+13]. This is motivated by the idea that a strong density oscillation in a chromophore's associated space can be identified with its energy. In which sense the latter is connected to an energy or an energy density is again not directly clear.

In order to shed some light upon this problem and present ideas of meaningful quantities that can be interpreted as energy densities or energies per chromophore, I take a step back and discuss a simple two-level donor-acceptor model. In particular, I want to model the density fluctuation one expects after a boost-like excitation of the donor. This is done in the following section.

5.4.1. A two-level donor-acceptor model

Model Hamiltonian As a first step I discuss the eigenvalues and eigenfunctions of a coupled two-level system consisting of a donor (D) and an acceptor (A). I model this system in the basis of $|\mathrm{AD}\rangle$ (ground state), $|\mathrm{A^*D}\rangle$ (only A is excited), and $|\mathrm{AD^*}\rangle$ (only D is excited) by the exciton Hamiltonian

$$\begin{aligned}\hat{H}^{\mathrm{ex}} = {} & E_{\mathrm{AD}}|\mathrm{AD}\rangle\langle\mathrm{AD}| \\ & + E_{\mathrm{A^*D}}|\mathrm{A^*D}\rangle\langle\mathrm{A^*D}| + E_{\mathrm{AD^*}}|\mathrm{AD^*}\rangle\langle\mathrm{AD^*}| \\ & + V|\mathrm{A^*D}\rangle\langle\mathrm{AD^*}| + V|\mathrm{AD^*}\rangle\langle\mathrm{A^*D}|\,.\end{aligned} \tag{5.3}$$

E_{AD}, $E_{\mathrm{A^*D}}$, and $E_{\mathrm{AD^*}}$ are the energies of the antisymmetrized product states $|\mathrm{AD}\rangle$, $|\mathrm{A^*D}\rangle$, and $|\mathrm{AD^*}\rangle$. V is the coupling matrix element $\langle\mathrm{A^*D}|\hat{V}^C|\mathrm{AD^*}\rangle = \langle\mathrm{AD^*}|\hat{V}^C|\mathrm{A^*D}\rangle$ with the Coulomb interaction $\hat{V}^C$, which mediates between the excited donor and acceptor states. In many typical exciton models [CGK06, §9.2], the ground state is not necessarily part of the model Hamiltonian but the system is only modeled in the vector space that is spanned by $|\mathrm{A^*D}\rangle$ and $|\mathrm{AD^*}\rangle$. Since I want to model the density oscillation that results from a real-time TDDFT calculation after a boost-like excitation, I include the ground state explicitly into the model.

A resonant coupling in the following means that $E_{\mathrm{A^*D}} = E_{\mathrm{AD^*}}$. Otherwise, donor and acceptor are non-resonant. I further define

$$\bar{E} = \frac{E_{\mathrm{A^*D}} + E_{\mathrm{AD^*}}}{2} \quad \text{and} \quad \Delta E = \frac{E_{\mathrm{A^*D}} - E_{\mathrm{AD^*}}}{2}\,. \tag{5.4}$$

The eigenvalues of $\hat{H}^{\mathrm{ex}}$ are $E_0 = E_{\mathrm{AD}}$ and $E_{1/2} = \bar{E} \pm \sqrt{\Delta E^2 + V^2}$ with corresponding eigenvectors

$$\begin{aligned} |0\rangle =& |\mathrm{AD}\rangle \\ |1/2\rangle =& \pm \operatorname{sgn}(V) \frac{1}{\sqrt{2}} \sqrt{1 \pm \frac{\Delta E}{\sqrt{\Delta E^2 + V^2}}} |\mathrm{A^*D}\rangle \\ &+ \frac{1}{\sqrt{2}} \sqrt{1 \mp \frac{\Delta E}{\sqrt{\Delta E^2 + V^2}}} |\mathrm{AD^*}\rangle\,, \end{aligned} \tag{5.5}$$

where $\operatorname{sgn}(V) = 1(-1)$ for $V \geq 0$ ($V < 0$). In this notation the upper sign belongs to the first state $|1\rangle$ and the lower sign belongs to the second state $|2\rangle$. In the case of resonant coupling with $\Delta E = 0$, the excited eigenstates reduce to a symmetric and an antisymmetric coupling between the single acceptor and donor transitions $|1/2\rangle = \frac{1}{\sqrt{2}}(\pm\operatorname{sgn}(V)|\mathrm{A^*D}\rangle + |\mathrm{AD^*}\rangle)$ with energies $E_{1/2} = \bar{E} \pm |V|$. In the case of vanishing coupling with $V = 0$ and $\Delta E > 0$, the eigenvectors are $|1\rangle = |\mathrm{A^*D}\rangle$ and $|2\rangle = |\mathrm{AD^*}\rangle$ with energies $E_1 = E_{\mathrm{A^*D}}$ and $E_2 = E_{\mathrm{AD^*}}$. For $\Delta E < 0$ the eigenvectors are $|1\rangle = |\mathrm{AD^*}\rangle$ and $|2\rangle = -|\mathrm{A^*D}\rangle$ with energies $E_1 = E_{\mathrm{AD^*}}$ and $E_2 = E_{\mathrm{A^*D}}$.

The splitting of the energy levels of the single chromophores into the energy levels of the coupled exciton system is often called Davydov splitting [Dav64]. In a real-time TDDFT simulation, the dipole moment shows oscillations with the excitation energies $E_{1/2} - E_0$ as discussed in section 4.2. Therefore, the dipole spectrum shows two peaks that are separated by $2\sqrt{\Delta E^2 + V^2}$.

This can be used to investigate coupling strengths between chromophores from a real-time TDDFT calculation. If one of the excited eigenstates is dipole-forbidden, such as the antisymmetric state in the case of resonant coupling between two equal chromophores, one must choose an excitation and an observable that do not show this dipole symmetry in order to see the spectral line. One option is to apply the boost in the donor half-space only and observe the dipole moment in the donor or acceptor half-spaces separately. I use this to compute the coupling strengths between two resonant sodium dimers and two BChls for different distances in a subsequent section. In the time domain, the dipole moments in the donor and acceptor half-spaces show a beat signal that can be evaluated equivalently for the coupling strength [HKK10].

Boost excitation For simplicity, I assume in the following that the single donor and acceptor states do not overlap and the space can be divided into a donor half-space V_{D} and an acceptor half-space V_{A}. In order to model the density fluctuation that arises from a real-time TDDFT calculation, I apply a weak boost-like excitation in the donor half-space V_{D} at $t = 0$. After that, the system is in the state

$$\begin{aligned} |\psi(t=0)\rangle &= c_{\mathrm{AD}}|\mathrm{AD}\rangle + c_{\mathrm{AD^*}}|\mathrm{AD^*}\rangle \\ &= c_0|0\rangle + c_1|1\rangle + c_2|2\rangle\,. \end{aligned} \tag{5.6}$$

The boost is applied in the way of section 4.2. Hence, $c_{\mathrm{AD}^*} = i\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}$ with the transition dipole $-e\boldsymbol{r}_{\mathrm{DD}^*}$ of the $\mathrm{D}\to\mathrm{D}^*$ excitation and $c_{\mathrm{AD}} = \sqrt{1-|c_{\mathrm{AD}^*}|^2}$ due to the normalization of the states. The coefficients $c_j = \langle j|\psi(t=0)\rangle$, which relate to the eigenvectors in the second line of equation (5.6), are calculated by expressing the states $|\mathrm{AD}^*\rangle$ and $|\mathrm{AD}\rangle$ through the eigenvectors. The coefficients are

$$\begin{aligned} c_0 &= c_{\mathrm{AD}} = \sqrt{1-|\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}|^2} \\ c_{1/2} &= c_{\mathrm{AD}^*}\frac{1}{\sqrt{2}}\sqrt{1\mp\frac{\Delta E}{\sqrt{\Delta E^2+V^2}}} = i\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}\frac{1}{\sqrt{2}}\sqrt{1\mp\frac{\Delta E}{\sqrt{\Delta E^2+V^2}}} \end{aligned} \tag{5.7}$$

The state of the system for $t>0$ is

$$|\psi(t>0)\rangle = \sum_{j=0}^{2} c_j|j\rangle e^{-i\omega_j t} \quad \text{with} \quad \hbar\omega_j = E_j\,. \tag{5.8}$$

Note that the coefficients c_0, c_1, and c_2 are time-independent and given by equation (5.7) since they measure the occupation of the eigenstates $|0\rangle$, $|1\rangle$, and $|2\rangle$ after the boost. Since the Hamiltonian remains time-independent for $t>0$, the occupation of the eigenstates remains constant as well. The coefficients $c_{\mathrm{A}^*\mathrm{D}}$ and c_{AD^*}, however, are time-dependent. If donor and acceptor states are non-overlapping, the coefficients are given through

$$\begin{aligned} c_{\mathrm{A}^*\mathrm{D}}(t) &= \langle \mathrm{A}^*\mathrm{D}|\psi(t)\rangle \\ &= i\frac{\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}}{2}\frac{V}{\sqrt{\Delta E^2+V^2}}\left[e^{-i\omega_1 t}-e^{-i\omega_2 t}\right] \end{aligned} \tag{5.9}$$

$$\begin{aligned} c_{\mathrm{AD}^*}(t) &= \langle \mathrm{AD}^*|\psi(t)\rangle \\ &= i\frac{\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}}{2}\left[\left(1-\frac{\Delta E}{\sqrt{\Delta E^2+V^2}}\right)e^{-i\omega_1 t}+\left(1+\frac{\Delta E}{\sqrt{\Delta E^2+V^2}}\right)e^{-i\omega_2 t}\right]. \end{aligned} \tag{5.10}$$

Hence, the probability of finding the system in one of the states $|\mathrm{A}^*\mathrm{D}\rangle$ or $|\mathrm{AD}^*\rangle$ is

$$|c_{\mathrm{A}^*\mathrm{D}}(t)|^2 = |\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}|^2\frac{V^2}{\Delta E^2+V^2}\sin^2(\tilde{\omega}t) \tag{5.11}$$

$$\begin{aligned} |c_{\mathrm{AD}^*}(t)|^2 &= |\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}|^2\left[\frac{\Delta E^2}{\Delta E^2+V^2}+\frac{V^2}{\Delta E^2+V^2}\cos^2(\tilde{\omega}t)\right] \\ &= |\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}|^2\left[1-\frac{V^2}{\Delta E^2+V^2}\sin^2(\tilde{\omega}t)\right] \end{aligned} \tag{5.12}$$

with $\hbar\tilde{\omega} = \sqrt{\Delta E^2+V^2} = \frac{1}{2}(E_1-E_2)$. This can be interpreted as an oscillation of the total excitation energy between donor and acceptor with frequency $\tilde{\omega}$. In the case of resonant coupling, this is a complete beat signal. Without coupling the energy remains at the donor. Finally, in the general case of $V\neq 0$ and $\Delta E\neq 0$, the beat is incomplete and only a part of the energy is transferred between donor and acceptor.

Density fluctuation By applying the same approximations for a weak boost as in section 4.2, one can write the density fluctuation as

$$\delta n(\boldsymbol{r},t) = \langle\psi(t)|\hat{n}|\psi(t)\rangle - n_0(\boldsymbol{r}) = -2i\sum_{j=1,2} c_j\sin(\omega_{0j}t)\rho_{0j}(\boldsymbol{r})\,. \tag{5.13}$$

Here, the ground-state density is $n_0 = \langle \mathrm{AD}|\hat{n}|\mathrm{AD}\rangle$. Following the same notation as in the preceding sections, $\hbar\omega_{0j} = E_j - E_0$ is the excitation energy for the $|0\rangle \to |j\rangle$ transition with transition density $\rho_{0j} = \langle 0|\hat{n}|j\rangle$.

One can express the transition densities in equation (5.13) through the basis vectors $\{|\mathrm{AD}\rangle, |\mathrm{A^*D}\rangle, |\mathrm{AD^*}\rangle\}$ by using equation (5.5). In the case of non-overlapping donor and acceptor states, $\langle \mathrm{AD}|\hat{n}|\mathrm{A^*D}\rangle = \rho_{\mathrm{AA}^*}$ and $\langle \mathrm{AD}|\hat{n}|\mathrm{AD^*}\rangle = \rho_{\mathrm{DD}^*}$ are the transition densities of the single $\mathrm{A} \to \mathrm{A}^*$ and $\mathrm{D} \to \mathrm{D}^*$ transitions of acceptor and donor, respectively. This results in

$$\rho_{01/02}(\boldsymbol{r}) = \pm\mathrm{sgn}(V)\frac{1}{\sqrt{2}}\sqrt{1 \pm \frac{\Delta E}{\sqrt{\Delta E^2 + V^2}}}\rho_{\mathrm{AA}^*}(\boldsymbol{r}) + \frac{1}{\sqrt{2}}\sqrt{1 \mp \frac{\Delta E}{\sqrt{\Delta E^2 + V^2}}}\rho_{\mathrm{DD}^*}(\boldsymbol{r})\ . \tag{5.14}$$

By inserting equation (5.14) into equation (5.13), the density fluctuation can be expressed in terms of the donor and acceptor transition densities

$$\begin{aligned}\delta n(\boldsymbol{r},t) = \quad & \underbrace{2\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}\left\{\frac{V}{\hbar\tilde{\omega}}\left[\sin(\tilde{\omega}t)\cos(\bar{\omega}t)\right]\rho_{\mathrm{AA}^*}(\boldsymbol{r})\right\}}_{=\delta n_{\mathrm{A}}(\boldsymbol{r},t)} \\ & + \underbrace{2\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}\left\{\left[\cos(\tilde{\omega}t)\sin(\bar{\omega}t) - \frac{\Delta E}{\hbar\tilde{\omega}}\sin(\tilde{\omega}t)\cos(\bar{\omega}t)\right]\rho_{\mathrm{DD}^*}(\boldsymbol{r})\right\}}_{=\delta n_{\mathrm{D}}(\boldsymbol{r},t)}\end{aligned} \tag{5.15}$$

with $\hbar\tilde{\omega} = \sqrt{\Delta E^2 + V^2}$ and $\hbar\bar{\omega} = \bar{E} - E_0$.

The density fluctuation in equation (5.15) has contributions from the single donor and acceptor transition densities, which can be assigned to density fluctuations in their respective half-spaces V_{D} and V_{A}. In the case of resonant coupling, it follows that $\frac{|V|}{\hbar\tilde{\omega}} = 1$ and $\frac{\Delta E}{\hbar\tilde{\omega}} = 0$. The respective expression for the density fluctuation

$$\begin{aligned}\delta n^{\Delta E=0}(\boldsymbol{r},t) = 2\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{D}}\{&\mathrm{sgn}(V)\left[\sin(\tilde{\omega}t)\cos(\bar{\omega}t)\right]\rho_{\mathrm{AA}^*}(\boldsymbol{r}) \\ &+ \left[\cos(\tilde{\omega}t)\sin(\bar{\omega}t)\right]\rho_{\mathrm{DD}^*}(\boldsymbol{r})\}\end{aligned} \tag{5.16}$$

has a symmetric form for acceptor and donor. On both sides the density has a rapidly fluctuating contribution $\bar{\omega}$ that is enveloped by a sine or cosine with frequency $\tilde{\omega}$. Again, the slow oscillation with $\tilde{\omega}$ can be identified with the energy transfer between donor and acceptor. The fast oscillation with $\bar{\omega}$ is the result of the ground state $|0\rangle = |AD\rangle$ being still occupied after the boost excitation.

In the case of no coupling with $V = 0$, the situation is reversed and the density fluctuation reduces to

$$\begin{aligned}\delta n^{V=0}(\boldsymbol{r},t) &= 2\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{D}}\left\{\left[\cos(\tilde{\omega}t)\sin(\bar{\omega}t)\right] - \left[\sin(\tilde{\omega}t)\cos(\bar{\omega}t)\right]\right\}\rho_{\mathrm{DD}^*}(\boldsymbol{r}) \\ &= 2\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{D}}\sin\left(\frac{E_{\mathrm{AD}^*} - E_{\mathrm{AD}}}{\hbar}t\right)\rho_{\mathrm{DD}^*}(\boldsymbol{r})\ .\end{aligned} \tag{5.17}$$

As expected, only the initially excited donor density oscillates with its related excitation energy $E_{\mathrm{AD}^*} - E_{\mathrm{AD}}$.

In the general case for $\Delta E \neq 0$ and $V \neq 0$ in equation (5.15), the time dependence on the acceptor side remains simple with the amplitude of the fluctuation scaled down

by a factor of $\frac{|V|}{\sqrt{\Delta E^2+V^2}}$. On the donor side the time dependence gets more involved and is discussed in a subsequent section in more detail.

The time-dependent induced dipole moments of acceptor and donor, $\boldsymbol{\delta\mu}_\mathrm{A}$ and $\boldsymbol{\delta\mu}_\mathrm{D}$, follow directly from equation (5.15)

$$\begin{aligned}\boldsymbol{\delta\mu}(t) = &\underbrace{-2e\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}\left\{\frac{V}{\hbar\tilde{\omega}}\left[\sin\left(\tilde{\omega}t\right)\cos\left(\bar{\omega}t\right)\right]\boldsymbol{r}_{\mathrm{AA}^*}\right\}}_{=\boldsymbol{\delta\mu}_\mathrm{A}(t)} \\ &\underbrace{-2e\boldsymbol{k}\cdot\boldsymbol{r}_{\mathrm{DD}^*}\left\{\left[\cos\left(\tilde{\omega}t\right)\sin\left(\bar{\omega}t\right)-\frac{\Delta E}{\hbar\tilde{\omega}}\sin\left(\tilde{\omega}t\right)\cos\left(\bar{\omega}t\right)\right]\boldsymbol{r}_{\mathrm{DD}^*}\right\}}_{=\boldsymbol{\delta\mu}_\mathrm{D}(t)}\end{aligned} \tag{5.18}$$

with the transition dipoles

$$\boldsymbol{r}_{\mathrm{AA}^*}=\int_{V_\mathrm{A}}\boldsymbol{r}\rho_{\mathrm{AA}^*}(\boldsymbol{r})d^3r \quad\text{and}\quad \boldsymbol{r}_{\mathrm{DD}^*}=\int_{V_\mathrm{D}}\boldsymbol{r}\rho_{\mathrm{DD}^*}(\boldsymbol{r})d^3r\,. \tag{5.19}$$

The dipole moments of donor and acceptor show the same time dependence as the induced density.

5.4.2. Description of excitation-energy transfer

The amplitude of the density fluctuation or the induced dipole moment oscillates between donor and acceptor with the same frequency $\hbar\tilde{\omega}=\sqrt{\Delta E^2+V^2}$ as the occupation of the single donor and acceptor excited states from equations (5.11) and (5.12). It can therefore be used as an indicator for a total excitation energy per chromophore within the scope of the model.

I exemplify this by means of two sodium dimers that are aligned parallel to each other with an inter-dimer distance of $16\,\mathrm{a_0}$. The single dimers are aligned with their symmetry axis in z-direction and are separated from each other in x-direction. In one system the sodium dimers are resonant with an intra-dimer bond length of $5.78\,\mathrm{a_0}$ and a coupling strength[41] of $V=0.0695\,\mathrm{eV}$. In a second system the bond of the acceptor dimer is shortened by $0.5\,\mathrm{a_0}$ as in [Hof+13]. The latter results in an off-resonance[42] of $|\Delta E|=0.0628\,\mathrm{eV}$ and a coupling strength of $V=0.0661\,\mathrm{eV}$ with very similar values $|\Delta E|\approx V$.

A sodium dimer is very well described by the two-level model since it shows an isolated, strong excitation with transition dipole along its symmetry axis. I performed a real-time TDDFT simulation of the combined donor-acceptor systems with a boost in the donor half-space parallel to the symmetry axis of the sodium dimer with energy $E^\mathrm{boost}\approx 6.8\cdot10^{-4}\,\mathrm{eV}$. Figures 5.6 (a1) and (a2) show the resulting absolute density fluctuation[43] of acceptor and donor

$$\delta N_\mathrm{A/D}(t)=\int_{V_\mathrm{A/D}}|\delta n(\boldsymbol{r},t)|d^3r\,. \tag{5.20}$$

[41] The computation of coupling strengths is explained below.

[42] I computed ΔE from the excitation energies of the single dimers.

[43] Note that the integral over a transition density without taking its absolute value is always zero $\int\rho_{ij}d^3r=0$ since the underlying eigenstates $|i\rangle$ and $|j\rangle$ are orthogonal.

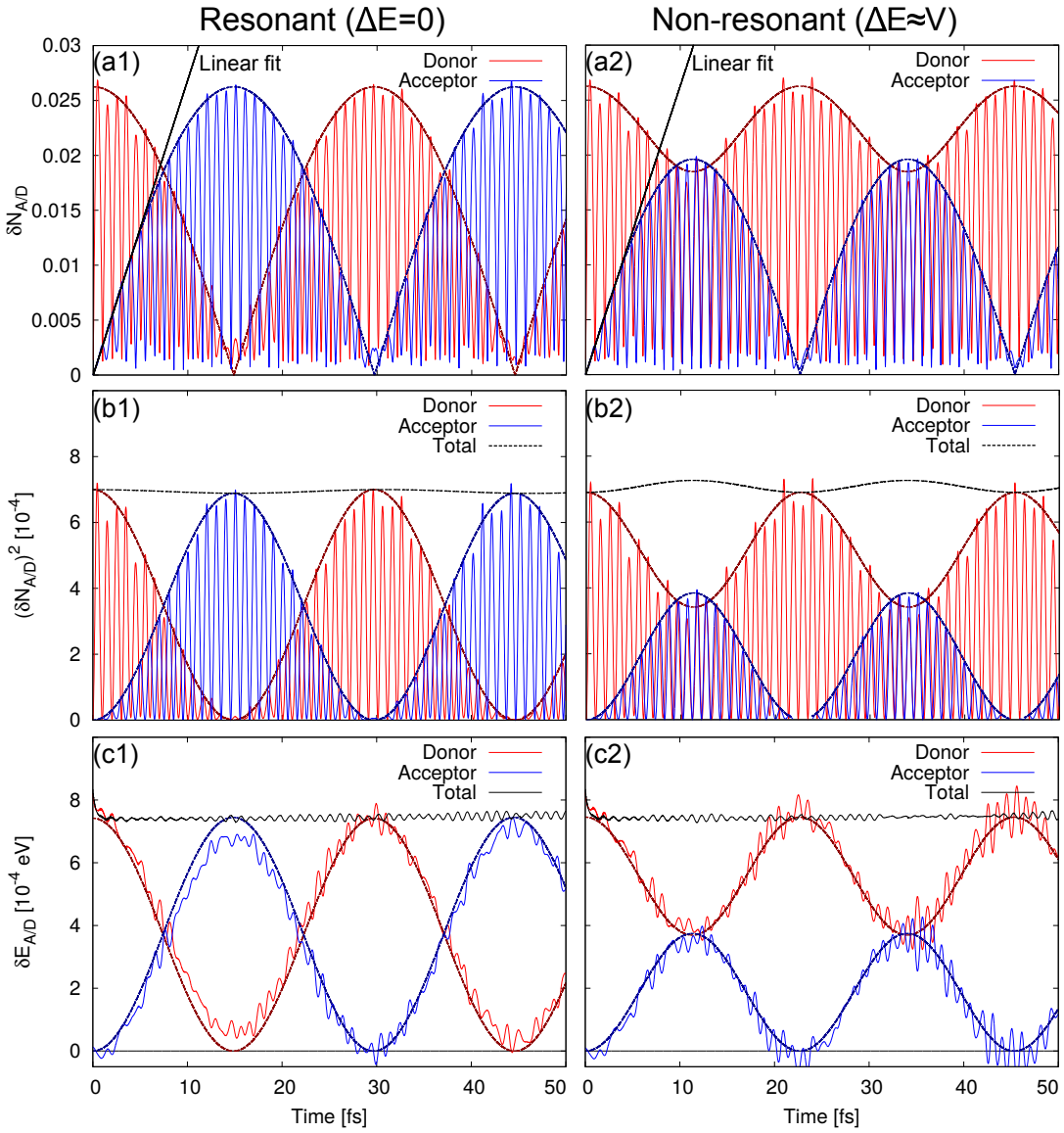


Figure 5.6: Absolute density fluctuation (a1)+(a2), squared absolute density fluctuation (b1)+(b2), and energy according to equation (5.2) (c1)+(c2) of donor and acceptor in a resonant (a1)-(c1) and a non-resonant (a2)-(c2) donor-acceptor system consisting of two sodium dimers. In figures (a1)+(a2) a linear fit to the envelope of the acceptor density fluctuation is displayed.

The envelope of $\delta N_{\mathrm{A/D}}$ is a measure for the strength of the density oscillation in the respective half-space. It can be estimated numerically by time-integration over the periodic time of the fast oscillation $\bar{\tau} = \frac{2\pi}{\bar{\omega}}$, i.e.,

$$\overline{\delta N}^{\bar{\tau}}_{\mathrm{A/D}}(t) \approx \frac{\pi}{2}\frac{1}{\bar{\tau}} \int_{t-\frac{\bar{\tau}}{2}}^{t+\frac{\bar{\tau}}{2}} \delta N_{\mathrm{A/D}}(t')dt' \quad \text{for} \quad \bar{\omega} \gg \tilde{\omega}\,. \tag{5.21}$$

The latter is usually fulfilled as long as the two chromophores can be described by the exciton Hamiltonian model and do not build a single molecule.

The envelopes of $\delta N_{\mathrm{A/D}}$ in figures 5.6 (a1) and (a2) are fits to the estimated envelopes from equation (5.21) with the assumed shape according to (5.15). In the resonant case in figure 5.6 (a1), δN_{A} and δN_{D} show the time dependence as predicted from the donor-acceptor model with the predicted beat. The envelope of the acceptor is proportional to $|\sin(\tilde{\omega}t)|$, the envelope of the donor is proportional to $|\cos(\tilde{\omega}t)|$ according to equation (5.16). In the non-resonant case in figure 5.6 (a2), the envelope of the acceptor shows the same behaviour proportional to $|\sin(\tilde{\omega}t)|$ as in the resonant case, just scaled by $\frac{|V|}{\sqrt{\Delta E^2+V^2}} \approx \frac{1}{\sqrt{2}}$ for $|\Delta E| \approx |V|$. The donor shows an envelope that can be described by $\sqrt{1-\frac{V^2}{\Delta E^2+V^2}\sin^2(\tilde{\omega}t)}$, which is discussed below.

The initial slope of the acceptor envelope is proportional to $|V|$ in both cases since in equation (5.15) $\left|\frac{V}{\hbar\tilde{\omega}}\sin(\tilde{\omega}t)\right| \overset{t\to 0}{\longrightarrow} \frac{|V|}{\hbar}t$. A linear fit to the acceptor envelope is also indicated in figures 5.6 (a1) and (a2). The initial slopes of the acceptor density fluctuation of both systems are, as expected, very similar since the coupling strength is almost identical. However, the beat frequency $\tilde{\omega}$ is larger by a factor of about $\sqrt{2}$ in the non-resonant case.

The envelope of $\delta N_{\mathrm{A/D}}$ already shows the correct frequency of the energy transfer between donor and acceptor. One expects the energy to behave in the way of the coefficients $|c_{A^*D}|^2$ and $|c_{AD^*}|^2$ from equations (5.11) and (5.12). This is perfectly reproduced by the energy per chromophore as defined through equations (5.1) and (5.2) and shown in figures 5.6 (c1) and (c2). As expected, the energy is completely transferred between donor and acceptor in the resonant case and only partially with amplitude $\frac{V^2}{\Delta E^2+V^2} \approx \frac{1}{2}$ in the non-resonant case. The total energy is indicated by the black line.[44]

The oscillations of the total energy mark the numerical accuracy. The faster oscillations that modulate the energies in the single donor and acceptor half-spaces cancel each other. They are no noise but probably the effect of slightly overlapping densities of donor and acceptor.

Considering the density fluctuations, the correct time dependence with the correct dependence on V and ΔE for the acceptor is achieved by the envelope of $(\delta N_{\mathrm{A}})^2$. In the resonant case, it is easily seen that this holds true for the donor. In the general case, the time-dependence of $(\delta N_{\mathrm{D}})^2$ according to (5.15) is more involved. However, one can show that

$$\left[\cos(\tilde{\omega}t)\sin(\bar{\omega}t) - \frac{\Delta E}{\hbar\tilde{\omega}}\sin(\tilde{\omega}t)\cos(\bar{\omega}t)\right]^2 = \left[1 - \frac{V^2}{\Delta E^2+V^2}\sin^2(\tilde{\omega}t)\right] h(t)\,, \tag{5.22}$$

[44] Note that the single contributions of the energy density as defined in equation (5.1) show rapid oscillations [Sol16] that cancel each other when summed up.

where $h(t)$ is a fast oscillating function with a time-dependent frequency but unity amplitude without beat. Hence, $\left[1 - \frac{V^2}{\Delta E^2+V^2}\sin^2(\tilde{\omega}t)\right]$ is the envelope of $(\delta N_\mathrm{D})^2$, which is the shape expected from the coefficient $|c_{\mathrm{AD}^*}|^2$. This also explains the shape of the envelope of δN_D in figure 5.6 (a2). The proof of equation (5.22) is presented in appendix D.

$(\delta N_\mathrm{A})^2$ and $(\delta N_\mathrm{D})^2$ are shown in figures 5.6 (b1) and (b2) for the resonant and non-resonant system. In the resonant case, both show exactly the behaviour one expects for an energy per chromophore, such as $\delta E_\mathrm{A/D}$ in figure 5.6 (c1). In the non-resonant case, the time-dependence for both, the donor and acceptor separately, is correct as well. However, the sum of the two envelops, indicated by the black line, is not constant.

The reason lies in the different amplitudes of the donor and acceptor oscillations since $\int |\rho_{\mathrm{AA}^*}| d^3r$ and $\int |\rho_{\mathrm{DD}^*}| d^3r$ in equation (5.15) are not necessarily equal. Since the non-resonant system still consists of two molecules that differ only slightly from each other, their transition densities are still similar. The interpretation of $(\delta N_\mathrm{A})^2$ and $(\delta N_\mathrm{D})^2$ as energy is therefore still valid. Yet, if the chromophores are very different, it is, to my knowledge, not guaranteed that $\int |\rho_{\mathrm{AA}^*}| d^3r$ and $\int |\rho_{\mathrm{DD}^*}| d^3r$ are similar. In this case, $(\delta N_\mathrm{A})^2$ and $(\delta N_\mathrm{D})^2$ cannot be analyzed quantitatively in this simple way.

Finally, I want to point out that in figures 5.6 (b1) and (b2) I show $\left[\int |\delta n_\mathrm{A/D}| d^3r\right]^2$, not $\int |\delta n_\mathrm{A/D}|^2 d^3r$. Still, I do not state that this is necessarily the best measure for a pragmatically defined energy per chromophore. There could be a possibility to relate the amplitude of the density fluctuation directly to an energy such that energy conservation is reproduced correctly. The numerical details of the presented calculations are listed in appendix E.2.3.

5.4.3. Prediction of coupling strengths

While the meaning of the density fluctuations as a direct measure for an energy per chromophore needs more investigation, one can utilize the results from the above sections to calculate coupling strengths and off-resonances from real-time TDDFT. This can be done in time space and frequency space in an equivalent way.

In time space one can get the values of V and ΔE from fits to the density fluctuations of donor and acceptor. This is equivalent to measuring the periodic time of the induced dipole moment [HKK10]. Since the initial slope of the acceptor density fluctuation is proportional to $|V|$, one can also use a linear fit to the envelope for small times to get relative values for $|V|$ for different distances.

In frequency space one can also utilize the induced acceptor and donor dipole moments from equation (5.18). This works for each component of the dipole moment as long as the excitation shows a dipole oscillation in this direction. The induced dipole moment is better suited for an analysis in frequency space since one does not need to take the absolute value of the density fluctuation and can simply use the results of section 4.2.

In the general case, one can write the time dependence of the acceptor dipole moment as

$$\frac{V}{\sqrt{\Delta E^2+V^2}}\left[\sin(\tilde{\omega}t)\cos(\bar{\omega}t)\right] = \frac{V}{\sqrt{\Delta E^2+V^2}}\frac{1}{2}[\sin[(\bar{\omega}+\tilde{\omega})t] - \sin[(\bar{\omega}-\tilde{\omega})t]]\,. \quad (5.23)$$

Thus, due to the results from section 4.2 the Fourier transformed dipole moment of the acceptor shows two sine-cardinal-shaped peaks at $\bar{\omega}+\tilde{\omega}$ with positive strength and at $\bar{\omega}-\tilde{\omega}$ with equal, negative strength (for $V > 0$ and reversed for $V < 0$).

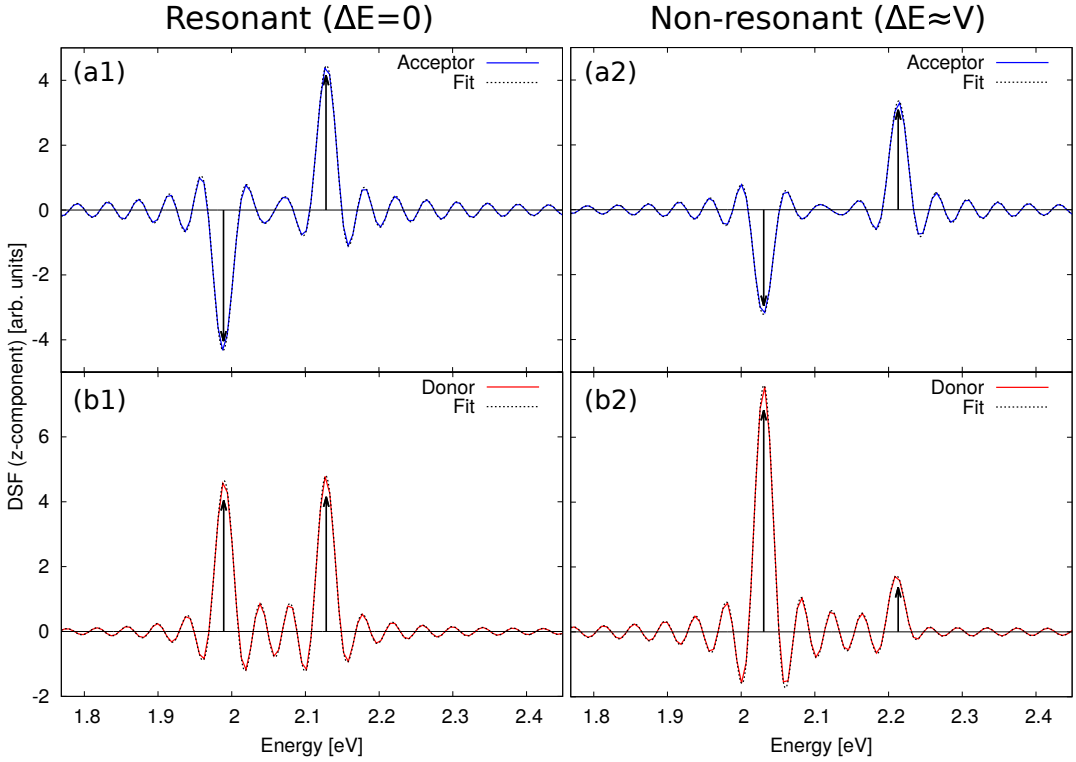


Figure 5.7: DSF as calculated from the z-component of the dipole moments of the acceptor (a1)+(a2) and donor (b1)+(b2) in the resonant (a1)+(b1) and non-resonant (a2)+(b2) sodium dimer systems. The single dimers are aligned with their symmetry axis parallel to the z-direction and are separated from each other along the x-direction.

This is shown in figures 5.7 (a1) and (a2) for the resonant and the non-resonant system, respectively. The perfect fit to the data (fitted spectral lines are indicated by the two arrows) indicates again that the time dependence of the density is well described by the donor-acceptor model.

The time dependence of the dipole moment of the donor can be written as

$$\left(1+\frac{\Delta E}{\sqrt{\Delta E^2+V^2}}\right)\sin[(\bar{\omega}-\tilde{\omega})t]+\left(1-\frac{\Delta E}{\sqrt{\Delta E^2+V^2}}\right)\sin[(\bar{\omega}+\tilde{\omega})t]\,. \tag{5.24}$$

In frequency space it also shows two peaks at the eigenenergies but both with positive heights since $\left|\frac{\Delta E}{\sqrt{\Delta E^2+V^2}}\right| \leq 1$.

This is shown in figures 5.7 (b1) and (b2) for the resonant and the non-resonant system. Note that in the non-resonant case, the excitation energy of the acceptor with the reduced bond length is larger than the energy of the donor excitation, i.e., $E_{\mathrm{A^*D}} > E_{\mathrm{AD^*}}$. This means that in this case $\Delta E > 0$ and the peak at $\hbar(\bar{\omega}-\tilde{\omega})$ is the stronger one.

The ratio between the heights of the donor lines is a direct measure for the off-resonance. To this end, I define $\bar{f} = f_+ + f_-$ and $\Delta f = f_+ - f_-$, where $f_+ \propto \left(1 - \frac{\Delta E}{\sqrt{\Delta E^2+V^2}}\right)$ and $f_- \propto \left(1 + \frac{\Delta E}{\sqrt{\Delta E^2+V^2}}\right)$ are the heights of the lines at $\bar{\omega} + \tilde{\omega}$ and $\bar{\omega} - \tilde{\omega}$. The ration between $|\Delta E|$ and $|V|$ results in $|\Delta E| = |V|\sqrt{\frac{\Delta f^2}{\bar{f}^2 - \Delta f^2}}$.

The evaluation in frequency space still works if $|V|$ is much smaller than the line width of a sine cardinal. In time space this means that the propagation time is shorter than the periodic time of the beat. In addition, the evaluation in frequency space still works if the system is not a proper two-level system. This is true as long as the coupling can be described as a coupling between two states, one from the donor and one from the acceptor.

In time space one cannot measure an accurate beat frequency or a periodic time in this case with a boost excitation. However, for the time-space evaluation one can apply a perturbation that mostly stimulates the excitations of interest, e.g., by an oscillating electric field that is tuned to the respective excitation energy. I use this approach to calculate coupling strengths between two resonant sodium dimers and two resonant BChls, respectively, for different distances in the following section.

5.5. Coupling strengths between chromophores from real-time TDDFT

Exciton Hamiltonians such as the one introduced in the last section are commonly used in the field of natural light harvesting systems [Hu+02; CGK06; SR06; JCM14]. The coupling strengths between the chromophores are often modeled by, e.g., Förster's dipole coupling or more involved approaches [För48; LKH99; CGK06; Hu+02]. In the following, I use real-time TDDFT with TDLDA to determine the coupling strength between two resonant sodium dimers and between the Q_y transitions of two resonant BChls. The sodium dimers are those already introduced in the last section with transition dipoles along the z-direction and separated along the x-axis. For the two BChls I used the structure from [ONS10] rotated by 45° around the z-axis and subsequently by −45° around the x-axis to orient the bacteriochlorin ring approximately parallel to the xy-plane. I duplicated this structure and separated the two resulting BChls along the z-axis to align them face-to-face.

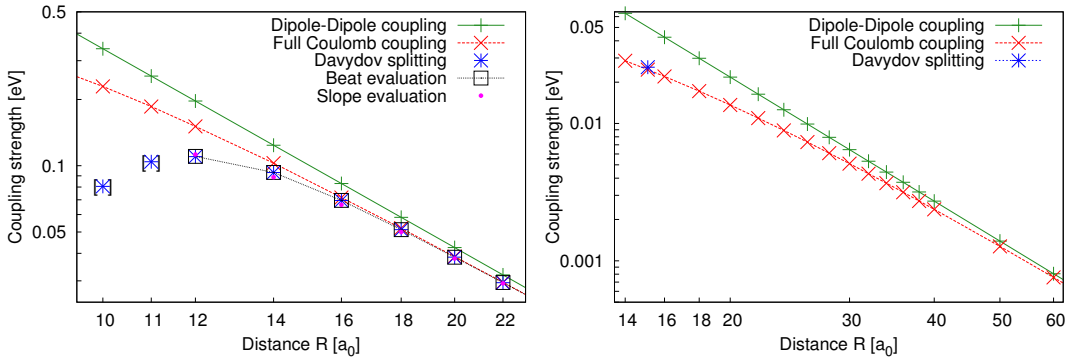
From real-time TDDFT I computed the coupling strengths V^{Dav} mainly in the frequency domain as described above. I compare the TDDFT results to the strengths from a pure Coulomb-like coupling between the chromophores. The full Coulomb coupling matrix element can be expressed through the transition densities ρ_{AA^*} and ρ_{DD^*} and reads [För48; SFK09; JCM14]

$$V^{\mathrm{Coul}} = \frac{e^2}{4\pi\varepsilon_0} \iint \frac{\rho_{\mathrm{AA}^*}(\boldsymbol{r})\,\rho_{\mathrm{DD}^*}(\boldsymbol{r'})}{|r - r'|} d^3r d^3r' \,. \tag{5.25}$$

V^{Coul} is often approximated through a Förster-type coupling between the transition dipoles [För48; SFK09; HKK10]

$$V^{\mathrm{dd}} = \kappa e^2 \frac{|\boldsymbol{r}_{\mathrm{AA}^*}||\boldsymbol{r}_{\mathrm{DD}^*}|}{4\pi\epsilon_0 R^3} \,. \tag{5.26}$$

$\kappa = \boldsymbol{e}_{\mathrm{DD}^*} \cdot \boldsymbol{e}_{\mathrm{AA}^*} - 3(\boldsymbol{e}_{\mathrm{DD}^*} \cdot \boldsymbol{e}_{\mathrm{R}})(\boldsymbol{e}_{\mathrm{AA}^*} \cdot \boldsymbol{e}_R)$. $\boldsymbol{e}_{\mathrm{AA}^*}$ and $\boldsymbol{e}_{\mathrm{DD}^*}$ are the unit vectors in the direction of the respective transition dipoles of donor and acceptor. $\boldsymbol{e}_R$ is the unit



(a) Two sodium dimers parallel to each other.

(b) Two BChls aligned face-to-face.

Figure 5.8: Coupling strengths from transition densities, transition dipoles, and Davydov splitting in frequency and time domains for two sodium dimers and two BChls against the inter-chromophore distance.

vector in the direction of the relative vector between the two chromophores, i.e., $\boldsymbol{e}_x$ for the sodium system and $\boldsymbol{e}_z$ for the BChl$a$ system. In the following, I calculated the transition dipoles from the transition densities through

$$\boldsymbol{r}_{0j} = \int \boldsymbol{r}\rho_{0j}(\boldsymbol{r})d^3r\,. \tag{5.27}$$

Sodium dimers The different coupling strengths for the sodium system are displayed in figure 5.8a against the inter-chromophore distance R on a logarithmic scale. Per construction, the orientation factor for this system is $\kappa = 1$. As expected, the dipole-dipole strength shows the R^{-3} behaviour from equation (5.26). The coupling strengths from dipole-dipole and transition density coupling approach each other for large distances per construction through equation (5.27). I tested this explicitly for distances up to $1000\,\mathrm{a_0}$ (not shown) to ensure the validity of the computation of V^{Coul}, which is described in appendix E.2.3 along with other numerical details.

For distances of less than $20\,\mathrm{a_0}$ the dipole-dipole coupling begins to differ significantly from the full Coulomb coupling. Figure 5.9 shows the relative error of the dipole-dipole coupling compared to the full Coulomb coupling. The error behaves as a power law[45] $\propto R^{-2}$ and deceeds the 10% threshold at distances $R \geq 20\,\mathrm{a_0}$.

The Davydov strengths in figure 5.8a fit perfectly to those of the pure Coulomb coupling for $R > 16\,\mathrm{a_0}$. I tested this explicitly up to $R = 50\,\mathrm{a_0}$. For $R = 16\,\mathrm{a_0}$, V^{Coul} differs from V^{Dav} by $\approx 3\%$ and by $\approx 10\%$ for $R = 14\,\mathrm{a_0}$. The sodium dimers can still be described through two coupled chromophores for $R < 16\,\mathrm{a_0}$ but the coupling strengths cannot be calculated as a pure Coulomb coupling between the unperturbed transition densities of the single chromophores.

Finally, for $R < 10\,\mathrm{a_0}$ the spectrum cannot be interpreted in the sense of the two-level donor-acceptor model and a Davydov evaluation is not possible. The deviations from the donor-acceptor model already begin at $R = 11\,\mathrm{a_0}$. They become visible

[45] The next higher multipole order neglected by the dipole-dipole coupling decays $\propto R^{-5}$ which explains, compared to the dipole-dipole strength $\propto R^{-3}$, the $\propto R^{-2}$ behaviour of the error.

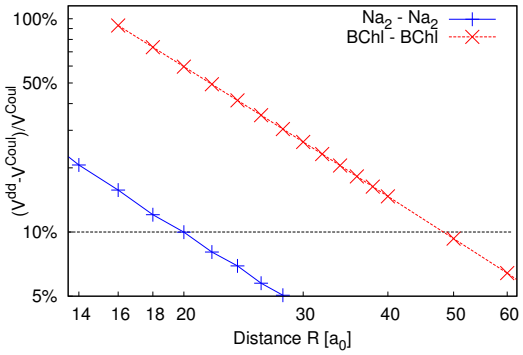


Figure 5.9: Error of the dipole-dipole coupling approximation compared to the full Coulomb coupling for different distances.

through the unequal heights of the two spectral lines and other excitations appearing at similar energies. The corresponding data points are enclosed in brackets in figure 5.8a.

In addition, I evaluated the coupling strength between the sodium dimers in the time domain by fitting the beat frequency to the envelope of δN_A (V^Beat) and fitting the slope (V^Slope) as shown in figure 5.6 (a1) and (a2). As expected, the beat-frequency evaluation in the time domain according to [HKK10] leads to the same results as the Davydov evaluation in the frequency domain. To get absolute values for the coupling strengths from the slope evaluation, I fixed the missing proportionality constant to the coupling strength from the beat-frequency evaluation at $50\,\mathrm{a_0}$, i.e., $V^\mathrm{Slope}(R = 50\,\mathrm{a_0}) \stackrel{!}{=} V^\mathrm{Beat}(R = 50\,\mathrm{a_0})$. The values of V^Slope shown in figure 5.8a are relative to this fixpoint and coincide well with the Davydov and beat-frequency evaluation. Still, for smaller distances the evaluation of the slope is less accurate since the beat becomes faster and the fitting range narrower.

Bacteriochlorophylls The coupling strength between the BChls for distances between $14\,\mathrm{a_0}$ and $60\,\mathrm{a_0}$ from dipole-dipole and full Coulomb coupling are shown in figure 5.8b. The orientation factor between the $\mathrm{Q_y}$ transition dipoles is $\kappa = 0.975842$ for this setup. The coupling strengths behave qualitatively similar to the ones of the sodium system but are smaller. The relative error of the dipole approximation compared to the full Coulomb coupling is also shown in Figure 5.9. Since the BChls are much larger than the sodium dimers, the error of the dipole approximation is much larger at the same distance. Still, it shows the same R^{-2} behaviour. The error of the dipole coupling reaches the 10% level at a distance of $R = 50\,\mathrm{a_0}$[46].

In addition to the discussed data, I computed a single coupling strength from the Davydov splitting approach for the same setup with an inter-BChl distance of $R = 8\,\text{Å} \approx 15.12\,\mathrm{a_0}$. The data point is marked in figure 5.8b. The predicted coupling strength of $V^\mathrm{Dav} = 0.02570\,\mathrm{eV}$ exceeds the one from pure Coulomb coupling $V^\mathrm{Coul} = 0.02469\,\mathrm{eV}$ by about 4 %. Hence, the Coulomb coupling as described by the unperturbed transition densities of the single BChls is still valid for this setup at

[46] Note that the calculation is done in vacuum.

$R \approx 15\,\mathrm{a_0}$. For smaller distances, one can again expect deviations between V^{Dav} and V^{Coul}. Numerical details can again be found in appendix E.2.3.

5.6. Conclusion and outlook

In this section I showed that the important Q-band transitions of single B850 BChls in the LH2 complex are well described by TDLDA as calculated from real-time and Q-Chem linear-response TDDFT and compared to Q-Chem ωPBE calculations. Still, Q-Chem TDLDA requires a large basis set for an accurate description and the TDLDA calculations in general show one or two additional, spurious states with a distinct charge-transfer character.

This becomes more disturbing in the TDLDA spectra of two coupled B850 BChls, which show a large number of spurious excitations. The latter partly result from a coupling between the $\mathrm{Q_y}$ and the spurious charge-transfer states of the single BChls and do not appear in the respective ωPBE calculation. The overall spectrum is not well described by TDLDA in this case.

Including the environment into the real-time TDLDA calculations resolves this issue and removes the most critical states. This opens the possibility to perform qualitatively reliable EET simulations within the B850 ring with real-time TDDFT in the future and to check the relevance of the spurious states in this respect.

For EET simulations one requires a meaningful measure for an energy density or an energy per chromophore. Apart from the energy density as defined in equation (5.1), the direct evaluation of the induced density fluctuation is a valid approach. In conclusion, real-time TDDFT provides an attractive tool for the efficient simulation of energy transfer processes in light-harvesting systems from first principles. Yet, to exploit its full power and flexibility to extract meaningful results requires more investigations, which is the subject of future work.

Appendix

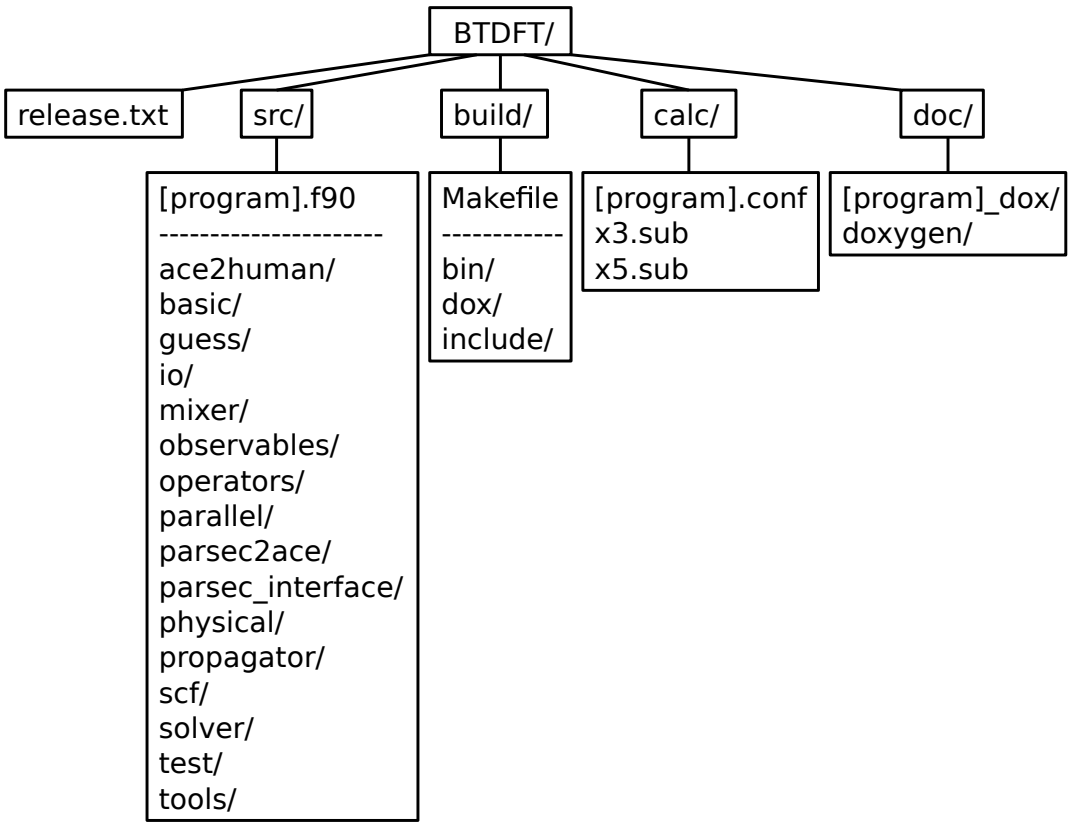


Figure A.1: BTDFT directory tree. [program] is a placeholder for any of the five BTDFT programs.

A. BTDFT - Additional documentation

A.1. File tree and release policy

A.1.1. The BTDFT file tree

The BTDFT directory tree is drawn in figure A.1. The contents of the directories are shortly described in the following. The source code files are listed in detail in the Doxygen documentation (see appendix A.5). Note also that BTDFT was previously referred to as yACES, which was its unofficial name.

release.txt The only file in the main project directory BTDFT/ is release.txt. For each release it contains the version tag and the change log.

src/ The directory src/ contains the source code of the BTDFT project. It contains the main files of the five BTDFT programs (BTDFT_td.f90, BTDFT_gs.f90, BTDFT_guess.f90, ace2human.f90, and parsec2ace.f90) and directories with further source files as listed in figure A.1.

build/ The build/ directory is used for compiling the BTDFT programs and the Doxygen documentation. This is described in more detail in appendix A.2. It contains

a makefile and several directories. The include/ directory contains make include files called 'make.[architecture]' in which environment and compiler variables can be set for each architecture. The doc/ directory contains the configuration files for the Doxygen documentation. After the compilation, a bin/ directory is created, which contains the executables. Moreover, directories are generated that contain the module and object files for the specific programs and architectures. These are reused in a new compilation on the same architecture such that only new or changed files need to be compiled.

calc/ The calc/ directory contains sample submit files for the UBT clusters btrzx3 (x3.sub) and btrzx5 (x5.sub) as well as sample configuration files for BTDFT_td, BTDFT_gs, and BTDFT_guess. The latter contain all possible configuration parameters, which are explained in the Doxygen documentation. The programs ace2human and parsec2ace only use command line arguments and do not need extra configuration files.

doc/ The doc/ directory contains one extra documentation folder for each of the five BTDFT programs with the ending '_dox'. They contain several files with the main documentation pages that are built into the Doxygen documentation. After compilation, a directory doxygen/ is generated that contains the html documentation for BTDFT. The documentation can then be opened by loading the respective doxygen/[program]/html/index.html file in a browser.

git There are hidden files and directories such as .git/ and .gitignore. They are described, as far as necessary, in appendix A.6.

A.1.2. Release history and release policy

The current release has the tag v1.6.5. The file release.txt in the main project directory contains the release history and the change log. The version number is generated as following:

- The third digit is increased if the new release only contains small changes of the existing code and bugfixes.
- The second digit is increased when new major features enter the code.
- The first digit is increased when backwards compatibility breaks, e.g., if the ACE file format changes.

A.2. Compilation and execution

Before compiling and running BTDFT, one should ensure that the right environment modules are loaded. The environment modules that are loaded at compile time should be the same as the ones that are loaded when running BTDFT.

A.2.1. Compilation

BTDFT The compilation of BTDFT is done in build/ in the main project directory. include/ contains files in which compiler, flags, library paths, preprocessor constants, etc. are specified. Currently, the libraries Intel MKL, gfortran, and the parallel version of ARPACK (PARPACK) are linked. PARPACK must be compiled separately.

The whole project is built by 'make'. The makefile in build/ tells 'make' to look for source code files in the src/ directory, compile them one after another, and finally link them. After compilation, an additional bin/ directory and several module and object directories are generated. bin/ contains the compiled executables with a suffix that identifies the respective architecture.

There is one module directory for each BTDFT program and for each architecture as well as one object directory for each architecture. The module and object files are reused if the project is built a second time on the same machine such that only new or changed files need to be compiled.

The architecture-specific module and object directories prevent conflicts if the same project directory is mounted in the file systems of different computers. Moreover, each of the BTDFT programs needs its own module directory and object file extension since the different programs partly use the same source code files but eventually a slightly different implementation. E.g., BTDFT_td and BTDFT_gs use the same routine to calculate the density from the KS orbitals. But, once the orbitals are real-valued and once they are complex-valued. The behaviour of the routine is controlled by preprocessor directives. Hence, a single source code file can result in different object and module files, depending on the program it is compiled for.

The targets in the makefile are.

'make' / 'make all' Build the five BTDFT programs.

'make prepare' and 'make header' Used internally to print information.

'make (program)' Only build the given program.

'make clean' Remove the binaries, module files, and object files for the current architecture.

'make clean_(program)' Clean up the files of a single program.

'make doc' Build the Doxygen documentation.

Doxygen documentation The Doxygen documentation is built in the build/ directory by calling 'make doc'. Each of the five BTDFT programs has its own documentation with an own Doxygen configuration file in build/dox/ with the suffix '*.cfg'. The most important parameters in the configuration file are those that set input and output directories, the project's name, a release tag, or a filter for input files. The second kind of file in 'build/dox/' has the suffix '.filter' (see appendix A.5 for an explanation).

Usually, the configuration files need not to be changed. Exceptions are, e.g., if a new release is built (then the new release tag must be set) or if a new source code file is not yet recognized by Doxygen due to some filter (then the new file has to be added to the list of input files).

ARPACK The ARPACK library [LSY97] and its parallel version PARPACK are required by BTDFT_gs to solve the ground-state KS equation. Precompiled versions for various machines, including the btrzx5 and btrzx3 clusters, exist. How to build PARPACK is explained in various README files in the corresponding directory and an additional file, which I wrote specifically for the UBT clusters.

A.2.2. Configuration

As most of the other contents, all configuration options are explained in the Doxygen documentation (see section A.5). There are three configuration files for each of the three programs BTDFT_guess, BTDFT_gs, and BTDFT_td. When running BTDFT, the configuration files must have the respective file names BTDFT_guess.conf, BTDFT_gs.conf, and BTDFT_td.conf. The order of parameters inside the files does not matter. Empty lines and comment lines starting with '#' can be added. The parameter keywords are case insensitive and additional delimiters such as '_' are ignored.

A.2.3. Submit files

The submit files are used for submitting a job to the clusters. Their main purpose is to request resources, set up the environment, and finally start the program. Sample submit files for btrzx5 and btrzx3 are included in the BTDFT/calc/ directory.

A.2.4. Execution

In the following, I explain how to set up a BTDFT calculation as well as the role of certain input and output files. This is done in the order the single programs are called one after another. I.e., an initial guess is created from the atom coordinates and the grid parameters. The initial state can be used to calculate the ground state, which can subsequently be used in a propagation. ace2human and parsec2ace are explained further below.

When running a program in a directory, all files that are required by that program have to be copied or linked into this directory. Symbolic links are often preferable for large input files, especially for the ACE files that are used as input to BTDFT_gs and BTDFT_td. Moreover, BTDFT_td and BTDFT_gs are the only programs that are parallelized. BTDFT_guess, ace2human, and parsec2ace are sequential programs.

BTDFT_guess A BTDFT calculation starts with an initial guess that is computed from BTDFT_guess. As input, BTDFT_guess requires the configuration file 'BTDFT_guess.conf' and one pseudo potential file for each atom type[47]. These must be called '[atom name]_POTRE.DAT'. The '[atom name]' is the usual atom abbreviation from the periodic table. The first letter must be upper case, the second letter, if it exists, is lower case.

[47] BTDFT usually uses non-local Troullier-Martins pseudo potentials [TM91] that are given on a radial, logarithmic grid. The Kleinman-Bylander transformation [KB82] is done inside the code. The local component of each atom specific pseudo potential must be chosen in the configuration file (see section 3.3.3).

The BTDFT_guess configuration file determines the grid size, grid spacing, atom types, their coordinates, and the local component of the atom specific pseudo potentials. From this, BTDFT_guess sets up the grid and calculates the initial density, the local ion potential, and the non-local pseudo potentials.

The initial guess is output as 'initial_guess.ace' and can be used as input to a ground-state calculation with BTDFT_gs. Moreover, a file 'guess.stat' is output that contains additional information about the initial guess.

BTDFT_gs The ground-state calculation requires an 'initial_guess.ace' file, the configuration file 'BTDFT_gs.conf' as input, and a submit file to run the job. The initial guess can be any valid ACE file but is usually taken directly from BTDFT_guess.

The configuration file determines, among others, the number of orbitals, the total charge of the system, if the calculation shall be spin polarized or spin unpolarized, which xc approximation shall be used, and numerical details such as convergence criteria and mixing parameters. The parallelization is set in the submit file by the requested resources and the number of MPI processes. At the current state BTDFT_gs only supports grid parallelization.

BTDFT_gs outputs a master output file 'master.out' with status information about the program setup and the single SCF iterations including the SRE error, KS eigenvalues, and energies. Additionally, each MPI process writes specific information into its own 'out.[process rank]' file.

At the end, one of 'ground_state.ace' or 'not_ground_state.ace', depending on the final state of convergence, and optional observable files are output. There is also an option to periodically output additional ACE files after a number of SCF iterations, which are called 'SCF_restart_[iteration].ace'. These can be used as initial guess to resume the ground-state calculation if, e.g., the calculation crashed for some reason.

BTDFT_td The propagation needs an 'initial_state.ace' file and the configuration file 'BTDFT_td.conf' as input and again a submit file. The initial state can be a ground state from BTDFT_gs or any other ACE file that contains orbitals. The latter excludes ACE files that contain an initial guess from BTDFT_guess and the restart files from BTDFT_gs. It is particularly possible to resume a propagation.

The configuration file determines the ratio between orbital and grid parallelization, propagation parameters such as the propagation time, the time step, and the propagator as well as external potentials, other kinds of excitations, and several observables. The total parallelization is again set in the submit file by the requested resources and the number of MPI processes.

BTDFT_td outputs a 'master.out' file and process specific 'out.[process rank]' files with similar contents as BTDFT_gs. At the end of the propagation, a 'final_state.ace' file is output, which contains the final state at the end of the propagation. This file can be used to resume the propagation. Of course, additional files are output that contain the requested observable data, etc.

ace2human ace2human is a command line program that is used to read and output the contents of ACE files. A complete list of all supported command line arguments is given in the corresponding Doxygen documentation.

Each time ace2human is called with a valid ACE file as input, a 'header.dat' and a 'reader.stat' are output. 'header.dat' contains all information in the header of the ACE file including electronic structure data and the grid parameters. 'reader.stat' only contains information about the success of the call. Additional command line arguments can be used to output the density, potentials, pseudo potentials, and orbitals as Gaussian cube files or as human-readable data files in a plane or parallel to any Cartesian axis.

parsec2ace parsec2ace is used to convert a special PARSEC [Kro+06] output file into a valid ACE file that can be used in a propagation. To this end, I extended the local version of PARSEC by an additional interface module (source code inside BTDFT/src/parsec_interface/). Since the grid parallelization of BTDFT differs from the one of PARSEC, parsec2ace offers the opportunity to rotate the system around the original coordinate axes of the PARSEC grid (subsequently up to two times around different axes). BTDFT works best if the largest half-axis of the boundary ellipsoid is aligned in z-direction. parsec2ace outputs a status file 'converter.stat' and the requested ACE file.

A.2.5. Practical remarks

At the end of this section I add some remarks about the choice of parameters and the parallelization.

System Alignment The parallelization is one-dimensional along the z-direction as introduced in section 3.3.1. For an optimal grid parallelization it is recommended to align the system with the largest half-axis of the boundary ellipsoid along the z-axis. This reduces the size of the halo layers and thus the MPI communication and memory access overhead when applying differential operators.

Grid size and grid spacing In view of the calculations presented in this work, the grid spacing was determined by the kind of atoms in the system. A grid spacing of $0.3\,\mathrm{a}_0$ is usually sufficient for systems that contain carbon, which can be seen from the discussion in appendix E.3.1. For sodium systems a larger grid spacing of $0.7\,\mathrm{a}_0 - 0.8\,\mathrm{a}_0$ is valid.

The grid size must be large enough that the multipole expansion that is used as boundary condition for the solution of Poisson's equation for the Hartree potential is valid. This depends on the size of the system, its shape, and its alignment inside the grid.

Time step The optimal time step depends on the propagator used and the time scales of the observed dynamics. The Taylor propagator is conditionally stable for time steps that are smaller than a critical time step, which depends on the grid spacing. For a grid spacing of $\Delta x = 0.3\,\mathrm{a}_0$, the critical time step is $\Delta t^{\mathrm{crit}} \approx 0.0005\,\mathrm{fs}$. Since the observed dynamics is usually much slower than this critical time step, there is no other restriction.

The Crank-Nicolson propagator is unconditionally stable. The reliability of the results depends on the validity of the time discretization. The time scales of the

dynamics is determined by, e.g., oscillating external potentials or the excitation frequencies. I found that a time step of up to $\Delta t^{\text{crit}} = 0.02\,\text{fs}$ is reliable to describe the dynamics in an energy range of up to $\approx 2-3\,\text{eV}$ (see section E.3.1). This corresponds to about 70-100 time steps per period of an oscillation with this energy. Excitations with higher excitation energies are typically red-shifted due to the discretization error.

A larger Crank-Nicolson time step requires more computation time than a Taylor time step. Yet, due to the larger possible time-step sizes it is typically faster. Moreover, the potentials are evaluated once per time step and therefore less frequent with a larger Crank-Nicolson time step.

Parallelization If finite differences of 6^{th} order are used, each MPI process should contain at least six xy-planes of grid points in its own range[48]. In this case, the halo region of each MPI process consists of three xy-planes below and above its own range. The halo layers occupy extra memory and must be loaded from the memory each time the finite differences are applied. Since the total halo region is as large as the process' own range (i.e., six xy-planes), the program requires about twice the memory of a sequential run and the performance is at best 50% of what would be possible with optimal scaling. A higher parallelization is still valid but uses disproportionate resources, also in view of the additional MPI communication.

In a propagation the orbital parallelization can be added in addition to the grid parallelization. The processes are organized in orbital units as explained in section 3.3.2 with grid parallelization within each orbital unit. The number of orbital units can be specified in the BTDFT_td configuration file.

There are two hard constraints to the number of orbital units: First, there must be at least as many orbitals as orbital units such that each orbital units contains at least one orbital. Second, the number of MPI processes must be an integer multiple of the number of orbital units (see figure 3.5).

A good practice is to choose a certain grid parallelization, e.g., among one or two nodes, and increase the number of requested nodes proportional to the number of orbital units. This way the second constraint is always fulfilled.

Additionally, the number of orbitals should be an integer multiple of the number of orbital units, if possible. The latter states that all orbital units contain the same number of orbitals, which ensures an optimal load balance.

PARPACK basis PARPACK builds up a $2N_{\text{orb}} + N_{\text{extra}}$-dimensional subspace of the N_{grid}-dimensional vector space and stores as many basis vectors. N_{orb} is the number of orbitals, N_{grid} is the number of grid points, and N_{extra} is an additional basis size that can be specified in the BTDFT_gs configuration file.

In general, N_{extra} should be larger the more orbitals are treated. If N_{extra} is too small, this can lead to convergence issues or poor performance. For small systems $5 \leq N_{\text{extra}} \leq 10$ is usually sufficient. For a ground-state calculation with 706 orbitals and $15.2 \cdot 10^6$ grid points I used up to $N_{\text{extra}} = 100$.

[48] In this consideration I ignore the ellipsoidal shape of the grid.

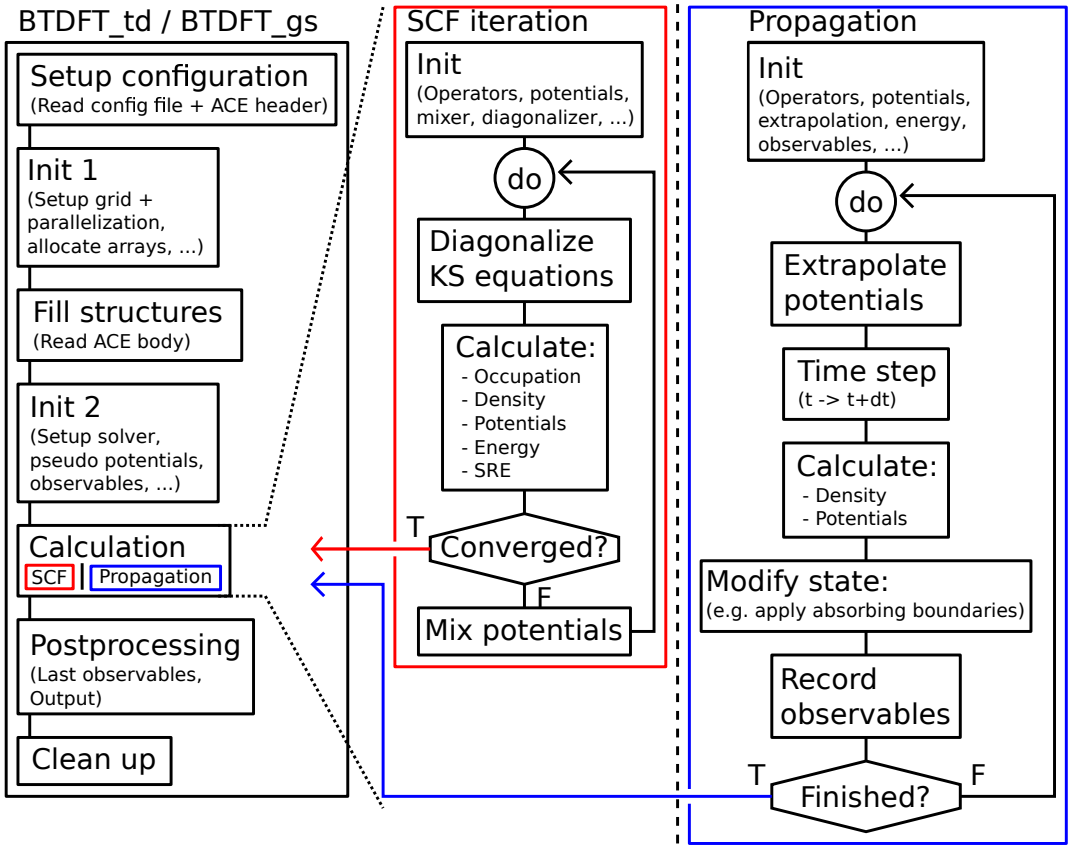


Figure A.2: Overview of the BTDFT_td and BTDFT_gs program sequences.

A.3. Implementation principles

A.3.1. Program sequences

An overview of the program sequences of BTDFT_td and BTDFT_gs is shown in figure A.2. The left part of the figure shows the contents of the main files BTDFT_td.f90 and BTDFT_gs.f90. In the sketch, they only differ in the 'Calculation' part when BTDFT_gs starts the SCF iteration whereas BTDFT_td starts the propagation. An overview of the sequence of the SCF iteration and the propagation are displayed in the middle and right parts of figure A.2. The single parts are discussed in the following:

Setup configuration: The main derived types, which contain data and subroutines concerning, e.g., the grid structures, the parallelization, and the electronic structure, are declared once at the beginning of BTDFT_gs and BTDFT_td. Both programs then read the contents of the respective configuration file (BTDFT*.conf) and the contents of the input ACE file. In BTDFT_gs the input ACE file must be called initial_guess.ace. In BTDFT_td is must be called initial_state.ace.

The parameters in the configuration file determine the parameters for the respective SCF iteration or propagation such as convergence criteria and the propagation time. The input ACE file contains an initial guess (BTDFT_gs) or an initial state (BTDFT_td). Its header contains information about the number of atoms and atom

types, the grid, etc. with a fixed layout.

Input parameters and ACE file header are then used to create configuration structures that are used to configure all major derived types in the 'Init 1' block. The compatibility of the ACE file header and the configuration parameters are checked. The configuration files are then broadcasted to all MPI processes.

Init 1: The ACE file header and the configuration parameters contain all information that is necessary to set up the major derived types. This means setting parameters in the respective data structures, setting up the parallelization and the grid, and allocating arrays for, e.g., potentials, density, and orbitals.

Fill structures: After all parameters are set and arrays are allocated the ACE file body is read into the structures. This is done in parallel by all MPI processes, which open the ACE file collectively using MPI_IO routines. Therefore, each MPI process reads its own data from the ACE file body.

For very large ACE files, which usually contain many KS orbitals with a large number of grid points, this has the special advantage that a single process does not need to store a complete set of data that must be scattered across all MPI processes. For really large systems, a single orbital can theoretically occupy more than 1 GB in the main memory, a complete ACE file many tens of GB. This can, in certain situations, lead to segmentation faults due to a lack of main memory capacity.

Init 2: The second initialization step completes the initialization. Here, all things are done that require the explicit knowledge of data such as the atom coordinates and the non-local parts of the pseudo potentials. Since the latter are only defined on subgrids around the atoms they can first be scattered across the MPI processes as soon as the coordinates of the atoms and the subgrids of the pseudo potentials are known. Since this information depends on the number of atoms and atom types, it is stored in the body part of the input ACE file and is therefore first available after the 'Fill structures' block in figure A.2 (see appendix A.4 for information about the format of the ACE file).

Calculation [SCF / Propagator]: In the 'Calculation' part the SCF iteration or the propagation are started, respectively. In the main program this is just a single subroutine call, which takes the initialized derived types such as the grid and the initial electronic structure data as input and returns the converged or propagated data. The flow charts of the SCF iteration and propagation are displayed in the middle and right parts of figure A.2 and explained in the following.

SCF: At the beginning of the SCF iteration, additional structures such as operators and the mixer are initialized and the potentials are calculated for the first time from the initial guess density. The self-consistency loop then starts with the diagonalization of the KS equations, which returns new KS orbitals and KS eigenvalues. From those the new occupation numbers, density, and potentials are calculated. From the new and old potentials the SRE error is determined and checked for convergence. If the iteration converged, the routine returns to

the main program. Otherwise, the new and old potentials are mixed and used for the next iteration cycle.

Propagator: As for the SCF iteration, additional structures are initialized and the potentials are calculated from the initial density. The loop performs time steps from $t \to t + \Delta t$ until the total propagation time is reached. Since the potentials at $t + \frac{\Delta t}{2}$ are used in a propagation step, the potentials are first extrapolated from t, $t - \Delta t$, $t - 2\Delta t$, ... to $t + \frac{\Delta t}{2}$ using a polynomial expansion (second order works well). The orbitals are then propagated from $t \to t + \Delta t$ using the Taylor of Crank-Nicolson propagator. From the new orbitals the density and potentials at $t + \Delta t$ are calculated and observables such as the dipole moment are recorded. In between, the state can be modified by, e.g., applying special absorbing boundaries as used for ARPES or charge-transfer simulations [Sch16; Dau16; Dau+16].

Postprocessing: After the SCF iteration or the propagation have finished and the converged or propagated data are returned from the 'Calculation' block, the 'Postprocessing' block calculates outstanding observables and outputs the respective data. In BTDFT_gs an ACE file named ground_state.ace or not_ground_state.ace is output, depending on the success of the SCF iteration. In BTDFT_td an ACE file named final_state.ace is output from which the propagation can be resumed.

Clean up: Finally, all main structures are destroyed, i.e., the underlying arrays are deallocated.

A.3.2. Grid details

Some operations are not orbital-specific. Examples are the calculation of the Hartree potential or the dipole moment, which only require the total density. In this case the operation is only grid-parallelized and all orbital units usually do exactly the same calculation.

In the case of a pure observable such as the dipole moment, it is sufficient if one orbital unit performs the computation. However, the computation can also be done in parallel by all processes if the grid is distributed among all processes, not only among the processes in one orbital unit. To achieve this, the processes in the same grid unit, which contain the same subgrids, must again split up this subgrid. This leads to new, smaller subgrids and new halo layers as well as halo communication within the grid units (if a semi-local operator is applied as for the Hartree potential).

This is clarified by figure A.3, which also illustrates the meaning of important index variables inside the *t_grid* structure of BTDFT. The figure shows 1D arrays on three processes, which are in the same grid unit, i.e., share the same subgrids but contain different orbitals. On the left-hand side and right-hand side of the figure, neighboring grid units are indicated. Also compare this figure to figures 3.3a on page 20 and 3.5 on page 22.

The single 1D arrays are visually divided into an upper and a lower section. The upper section shows the boundary (blue) and halo (red) layers in the orbital unit's

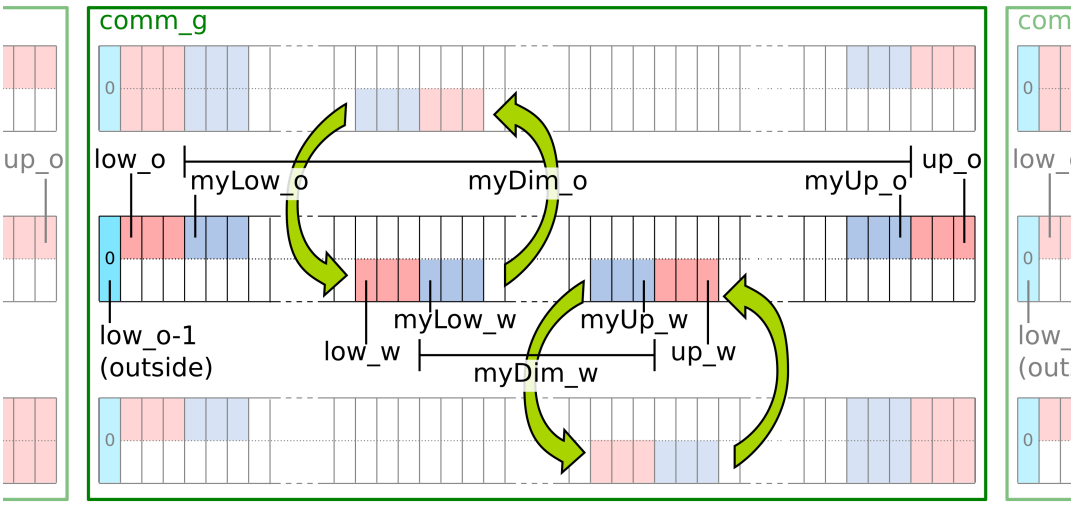


Figure A.3: Additional grid parallelization within a grid unit and important indices as found in the *t_grid* structure of BTDFT. Details are discussed in the text.

subgrid (or index range), which is labeled with *_o*. This orbital unit range is the one that is introduced in section 3.3.1.

The lower section of the 1D arrays shows boundary (blue) and halo (red) layers of the world communicator's range, which is labeled with *_w* and is not discussed in section 3.3.1. The latter is only relevant if the grid is parallelized among all processes. The green arrows indicate the halo communication inside the grid unit in the case that a semi-local operator is applied with grid parallelization among all processes in the world communicator.

An array that requires halo communication, such as the density or the KS orbitals, is allocated in its orbital unit range with halo layers and an outside element attached to a process' own range. Figure A.3 explains this for the middle process with orbital-unit specific (*_o*) quantities written above and world-communicator specific (*_w*) quantities written below the array. In view of this figure, the orbital-unit range runs from $low\_o - 1$ to up_o. The array element with one-dimensional index $low\_o - 1$ contains the outside-value 0, which is the value that is used for grid points outside the boundary ellipsoid. The process' own subgrid ranges from $myLow\_o$ to $myUp\_o$ and is the same on all processes in the same grid unit. The number of grid points in the orbital-unit range is $myDim\_o = myUp\_o - myLow\_o + 1$.

The additional grid parallelization within the grid unit leads to the world-communicator indices (*_w*), which differ between processes in the same grid unit. In this example with three orbital units each process in the grid unit gets one third of the grid unit's total subgrid. The meaning of the *_w* indices and $myDim\_w$ is the same as for the *_o* ones but with the total grid parallelization.

By default, arrays in BTDFT are allocated with their orbital unit range. The additional grid parallelization within the grid units can be useful in some situations, e.g., the calculation of the dipole moment. If an operation is performed by every orbital unit and grid-parallelized only within the respective orbital unit (*_o*) or grid-parallelized

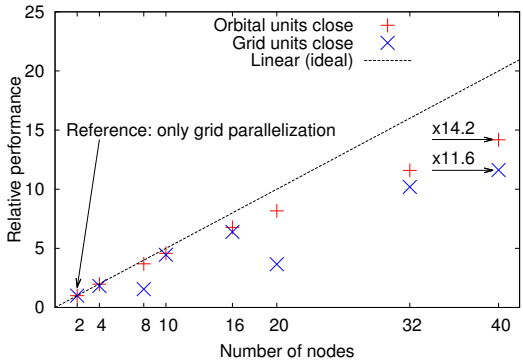


Figure A.4: Dependence of the performance of BTDFT for the polyacetylen chain from section 3.5.1 with different process mappings: Once the orbital units are close-packed, once they are scattered across all nodes.

among all processes ($\_w$) is typically controlled by a flag called *orbUnitFlag*. If this flag is *.true.*, which is the default, the computation is done by each orbital unit and only in the orbital-unit range.

The additional grid parallelization must be treated with care and does not generally lead to a better performance, especially with semi-local operators, which require halo communication. If the default grid parallelization with the orbital-unit ranges is already maxed out, the additional grid parallelization hampers the performance.

A.3.3. Mapping MPI processes onto the hardware

The 2D virtual topology, which BTDFT uses for its grid and orbital parallelization (see figure 3.5), must be mapped onto the hardware. As outlined in appendix A.2.5, it is rational to specify a grid parallelization among a certain number of nodes (M) and run the job with this number of nodes times the number of orbital units specified in the configuration file.

The virtual topology is set up such that the MPI processes in the same orbital unit are packed close on the hardware to achieve a good halo-communication performance. In this case, each orbital unit is only distributed among these M nodes. The MPI communication patterns are simple: Halo communication (or intra-orbital-unit communication) takes place within each set of M nodes and inter-orbital-unit communication between the sets of M nodes that contain different orbital units.

The opposite is also possible, i.e., scattering the orbital units across all nodes and pack the grid units close (but keeping the ratio of nodes per orbital unit). I compare the resulting performance of both approaches for the polyacetylen chain from section 3.5.1 in a similar way as in figure 3.11b. The results are displayed in figure A.4, again for a grid parallelization among two nodes (i.e., $M = 2$ in the close-packed case). Note that the performance cannot be compared directly to the one in figure 3.11b since I chose different numerical parameters and this version of BTDFT uses a predictor-corrector scheme instead of the potential extrapolation.

The close-packed variant shows a better performance and a more regular scaling behaviour while the scattered variant shows performance drops at 8 and 20 nodes. In

general, the communication paths with the scattered orbital units are very inhomogeneous and vary with different numbers of nodes if the ratio of nodes per orbital unit is kept constant. Therefore, the close-packed variant has clear advantages.

A.3.4. Optimized convergence criteria

The convergence criteria of the Crank-Nicolson and Hartree equations for the time-dependent code can be chosen automatically by BTDFT if a negative value is given as tolerance in the configuration file. However, this works only if the initial state is a proper ground state of the system.

Crank-Nicolson equation If one would start the propagation with the exact ground-state orbitals φ_j^{GS} and would not apply a perturbation, the time dependent KS orbitals would only get a phase factor $\varphi_j(t) = \varphi_j^{\mathrm{GS}} e^{-i\varepsilon_j t/\hbar}$. One can now take the approximate ground-state orbitals $\varphi_j^{\mathrm{GS,approx}}$ from the input and do a Crank-Nicolson time step Δt with the initial guess $\varphi_j^{\mathrm{GS,approx}} e^{-i\varepsilon_j \Delta t/\hbar}$ and a specified number of iterations. The final backward error of this procedure, or the minimum over all orbitals, is then taken as convergence criterion.

The argument behind this scheme is that it makes no sense to solve the Crank-Nicolson equation with a higher quality than the input orbitals yield. In practice, this worked fine so far, which I checked by choosing stronger and weaker criteria. A too weak criterion can be recognized by, e.g., line shapes in the dipole spectrum that are no proper sine cardinals. In the time-domain the dipole moment shows some self-enhancement or damping behaviour in this case. Still, if the input orbitals are of bad quality (or no ground-state orbitals), one should manually choose a criterion.

Poisson's equation for the Hartree potential To get a proper Hartree tolerance, BTDFT uses a similar scheme. This time, the input density is used to calculate the Hartree potential with the input Hartree potential as initial guess. This is again done with a specified number of iterations, usually one in this case. The final backward error is taken as convergence criterion. Since the input Hartree potential has been calculated before by the ground-state BTDFT code or PARSEC, the resulting criterion is a little stronger but very similar to that from the ground-state calculation.

A.3.5. File layout and implementation

The BTDFT project consists of several files including source code files, which are themselves organized in a directory tree. Nevertheless, most of the source code files follow a strict format and naming scheme, which is described in the following.

The main files of the five BTDFT programs are located in the src/ directory and called BTDFT_td.f90, BTDFT_gs.f90, BTDFT_guess.f90, ace2human.f90, and parsec2ace.f90. BTDFT strictly uses Fortran modules, which are included via the *use* statement, as well as derived data types and type bound procedures.

All module names begin with *m_* such as *m_grid* or *m_parallel*. The latter contain the data and routines concerning the real-space grid and the parallelization, respectively. Each Fortran module is implemented in a single file that is named after

the module such as grid.f90 and parallel.f90 in the upper example. In most of the modules a derived type is defined beginning with $t\_$ such as t_grid and $t\_parallel$. With this naming scheme it is always clear that, e.g., the grid data and routines are part of the t_grid structure in the m_grid module which is implemented in the file grid.f90.

```
module m_sample
![Use-includes ...]
implicit none

![Visibility]
private
public :: t_sample

![Derived types]
type :: t_sample
  integer, public :: ndim
  real, dimension(:), allocatable, public :: val
  ![Other data ...]
  contains
    procedure, public, pass :: init => sampleInitialize
    procedure, public, pass :: del => sampleDelete
    procedure, public, pass :: out => sampleOut
    ![Other type bound procedures ...]
end type

contains

![Module procedures]

!-----------------------------------------------------------------!
!Constructor
subroutine sampleInitialize(this,ndim)
implicit none
  class(t_sample) :: this
  integer, intent(in) :: ndim
  call this%del() !<=> call sampleDelete(this)
  this%ndim = ndim
  allocate(this%val(ndim))
  this%val = 0.0
end subroutine sampleInitialize

!-----------------------------------------------------------------!
!Destructor
subroutine sampleDelete(this)
implicit none
  class(t_sample) :: this
  if (allocated(this%val)) deallocate(this%val)
end subroutine sampleDelete

!-----------------------------------------------------------------!
!Print contents of the given t_sample structure 'this'
subroutine sampleOut(this,string)
implicit none
  class(t_sample) :: this
  character(*), intent(in) :: string
  write(*,*) string
```

```
52    write(*,*) this%ndim
53    write(*,*) this%val
54  end subroutine sampleOut
55
56  end module m_sample
```

Listing A.1: sample.f90

A typical file sample.f90 looks as displayed in listing A.1. Lines 7-8 in sample.f90 set the visibility of all module contents but the *t_sample* structure to private. E.g., if the module *m_sample* is use-included somewhere, only the *t_sample* structure is accessible from outside the module.

Lines 11-20 define the derived type *t_sample*, which contains a data part (lines 12-14) and type bound procedures (lines 16-19) after the types's *contains* statement (line 14). The type bound procedures are linked to module procedures that are implemented in the *m_sample* module (lines 25-54) after the module's *contains* statement (line 21). All components of *t_sample*, i.e., data and type bound procedures, are public and therefore accessible from outside the module through the *t_sample* structure using the % operator.

```
1  program useSample
2  use m_sample
3  ![Other use-includes ...]
4  implicit none
5
6  ![Declaration]
7  type(t_sample) :: sample, noSample
8
9  ![Initialization]
10 call sample%init(2) !<=> call sampleInitialize(sample,2)
11 call noSample%init(4) !<=> call sampleInitialize(noSample,4)
12
13 ![Output]
14 call sample%out("I'm a sample")
15 call noSample%out("I'm no sample!!")
16
17 ![Delete]
18 call sample%del()
19 call noSample%del()
20
21 end program useSample
22
23 ![Output]
24 ! I'm a sample
25 !            2
26 !    0.00000000        0.00000000
27 ! I'm no sample!!
28 !            4
29 !    0.00000000        0.00000000        0.00000000        0.00000000
```

Listing A.2: useSample.f90

The type bound procedures have the *pass* (default) or *nopass* attribute. The *pass* attribute states that the corresponding bound subroutine must get an instance of the structure itself as first argument, i.e., a variable of type *class*(*t_sample*) in this example. The latter is usually called *this* inside the module procedures.

A type bound procedure can be accessed through the structure via the % operator in the same way as any data inside the structure. When the subroutine is called as type bound procedure (with the *pass* attribute) of the corresponding structure, the first argument *this* is given implicitly (e.g., in the call of *sample%init* in useSample.f90). Two instances of *t_sample* are declared in useSample.f90 called *sample* and *noSample*, which both have their own data[49]. Hence, *sample* and *noSample* can be initialized independently from each other, contain different data, and produce different output. Still, they use the same implementation of the subroutines.

Allocatable arrays vs. Fortran pointers Inside the code I use allocatable arrays as well as Fortran pointers. To some extent these are interchangeable, yet they are different. I will briefly explain their differences and reasons for using one or the other in certain situations. Still, I restrict myself to the points that are important for this work.

Fortran pointers can be allocated as allocatable arrays, i.e., space in the main memory can be reserved. Still, in the code I only allocate allocatable arrays. Pointers, on the other hand, are not allocated in BTDFT but strictly used to point to arrays or other objects that already exist.

From my point of view, the main difference between Fortran pointers and allocatable arrays is that arrays are permanently assigned to the memory that was allocated for them whereas pointers can be assigned to other objects. This is illustrated in the code example below showing a sample file pointers.f90.

```
1  program pointers
2  use m_sample
3  implicit none
4
5  ![Declaration]
6  type(t_sample), target :: sample
7  type(t_sample), pointer :: p_sample
8  real, dimension(:), allocatable, target :: a1, a2
9  real, dimension(:), pointer :: p1, p2, ptmp
10
11 ![Allocate arrays]
12 allocate(a1(2))
13 allocate(a2(3))
14 a1 = 1.0
15 a2 = 2.0
16
17 ![Assign pointers]
18 p1 => a1
19 p2 => a2
20 write(*,*) p1
21 write(*,*) p2
22
23 ![Rotate pointers]
24 ptmp => p1
25 p1 => p2
26 p2 => ptmp
27 write(*,*) p1
28 write(*,*) p2
```

[49] Even arrays can be created from a derived type and derived types can be nested.

```
29
30 !-------------------------------------------------------------!
31
32 ![Initialize sample]
33 call sample%init(4)
34
35 ![Assign p_sample to sample]
36 p_sample => sample
37
38 ![Assign p1 to sample%val]
39 p1 => p_sample%val
40 write(*,*) p1
41
42 end program pointers
43
44 ![Output]
45 !Before rotation:
46 !    1.00000000        1.00000000                    (p1 points to a1)
47 !    2.00000000        2.00000000        2.00000000  (p2 points to a2)
48 !After rotation:
49 !    2.00000000        2.00000000        2.00000000  (p1 points to a2)
50 !    1.00000000        1.00000000                    (p2 points to a1)
51 !p1 points to p_sample%val (and hence to sample%val):
52 !    0.00000000        0.00000000        0.00000000        0.00000000
```

Listing A.3: pointers.f90

In listing A.3, $a1$ and $a2$ are allocatable arrays that are allocated and initialized inside the code (lines 12–15). The pointers $p1$ and $p2$ are assigned to $a1$ and $a2$ (lines 18–19), respectively, and can be used as aliases. The arrays $a1$ and $a2$ therefore need the *target* attribute. Fortran pointers are only references to a memory address. $p1$ and $a1$, at this point, are assigned to the same address and changing a value in $a1$ directly changes the same value in $p1$.

One example in which a pointer can be useful is in switching or rotating data between arrays (lines 24–26). After this rotation $p1$ is assigned to $a2$ and $p2$ is assigned to $a1$. When doing the same directly with the arrays $a1$ and $a2$ a lot of memory-to-memory copies must be performed, which is expensive in terms of computation time if the arrays are large. On the other hand, the same can also be done completely without $a1$ and $a2$ by directly allocating $p1$ and $p2$.

I do not use the direct allocation of pointers for a few reasons. First of all, if a pointer is allocated and subsequently assigned to another object, the originally allocated memory is not assigned nor accessible if not another pointer has been assigned to it before. Second, there is the threat of pointer aliasing that can in principle happen [HW10, §2.4.3]. Compilers are usually cautious and try to prevent this at compile time, which causes performance losses. Third, pointers can be assigned to non-contiguous arrays such as every second element of an array or a non-contiguous subarray of a multi-dimensional array. This may cause errors if the pointer is then given to a subroutine that expects a contiguous array.

Last but not least, pointers can also be assigned to derived types as with *p_sample* (line 36) or to components of derived types (line 39). In the latter case the whole derived type needs the *target* attribute.

Polymorphism The operator and solver structures in BTDFT use inheritance and polymorphism, which are concepts of object-oriented programming. I will therefore explain them briefly by means of a minimal example. Yet, the following is again far from being a complete reference.

The Krylov subspace solvers that are used in BTDFT only require the frequent application of a linear operator $\hat{O}$ to solve the equation $\hat{O}x = y$ for x. The operator needs not to be given explicitly as a matrix but only as a subroutine that takes some function (i.e., array) f as input and returns the result $\hat{O}f$. The Krylov subspace solver itself is generic and can be implemented without any knowledge of the operator.

To keep the solver implementation general, the solver shall get the inhomogeneity y and the operator $\hat{O}$ as derived type *t_operator* as input. A general operator structure must only implement a type bound procedure that knows how to apply the operator. The structure *t_operator* in listing A.4 below (lines 6–11) contains, in this minimal example, everything necessary: A type bound routine *apply* (line 10) and, since the operator acts on 1D arrays, the array dimension *ndim* (line 7).

```
1  module m_operator
2  ![Use-includes]
3  implicit none
4
5  ![Define abstract parent structure with deferred routine]
6  type, abstract :: t_operator
7    integer :: ndim
8    ![Further contents ...]
9    contains
10     procedure(apply), public, pass, deferred :: apply
11 end type
12
13 ![Define operator t_op1]
14 type, extends(t_operator) :: t_op1
15   ![Further contents ...]
16   contains
17     procedure, public, pass :: apply => op1Apply
18 end type
19
20 ![Define operator t_op2]
21 type, extends(t_operator) :: t_op2
22   ![Further contents ...]
23   contains
24     procedure, public, pass :: apply => op2Apply
25 end type
26
27 ![Abstract interface]
28 abstract interface
29   subroutine apply(this,i,o)
30     import :: t_operator
31     class(t_operator) :: this
32     integer, dimension(1:), intent(in) :: i
33     integer, dimension(1:), intent(out) :: o
34   end subroutine apply
35 end interface
36
37 contains
38
39 ![Module procedures]
```

```
40
41 ![t_op1 subroutine]
42 subroutine op1Apply(this,i,o)
43 implicit none
44   class(t_op1) :: this
45   integer, dimension(1:), intent(in) :: i
46   integer, dimension(1:), intent(out) :: o
47   if (size(i)/=this%ndim .or. size(o)/=this%ndim) return
48   o = i
49 end subroutine op1Apply
50
51 ![t_op2 subroutine]
52 subroutine op2Apply(this,i,o)
53 implicit none
54   class(t_op2) :: this
55   integer, dimension(1:), intent(in) :: i
56   integer, dimension(1:), intent(out) :: o
57   if (size(i)/=this%ndim .or. size(o)/=this%ndim) return
58   o = 2*i
59 end subroutine op2Apply
60
61 end module m_operator
```

Listing A.4: operator.f90

The operator structure is *abstract* (line 6) which states that no instance of *t_operator* can be created. The reason for this is that the *apply* procedure has no implementation, only an abstract interface (lines 28–35) that defines the layout of the *apply* routine. The interface tells the *t_operator* structure that its type bound *apply* procedure must get the the *t_operator* structure itself as well as an integer array i as input and returns an integer array o as output. Moreover, the *apply* procedure has the *deferred* attribute, which states that all derived types that *extend* the original, abstract operator type must implement the *apply* procedure. Therefore, *t_operator* is used as a template of an operator and defines a kind of scaffold.

In *m_operator*, there are two additional derived types called *t_op1* (lines 14–18) and *t_op2* (lines 21–25) that *extend* the abstract *t_operator*. Thus, they inherit the *ndim* component and the type bound *apply* procedure as well as the type itself. This means that *t_op1* and *t_op2* are both also of type *t_operator*. Since *t_op1* and *t_op2* are not *abstract*, they must implement the *apply* procedure with the correct interface (lines 28–35). The respective implementations are the module procedures *op1Apply* (lines 42–49) and *op2Apply* (lines 52–59).

The module procedure *solver* in listing A.5 simulates some kind of solver algorithm that solves $\hat{O}x = y$. The type of the input operator is *t_operator*. While *t_operator* itself is *abstract*, *t_op1* and *t_op2* are also of type *t_operator*. Thus, *t_op1* and *t_op2* are both valid inputs since both extent *t_operator* and therefore implement an *apply* procedure. Somewhere inside the solver algorithm the operator can then be applied (line 12).

The solver can be called as in listing A.6. There, the solver is called once with *t_op1* (line 21) and once with *t_op2* (line 25) and uses the corresponding implementation of *apply*, i.e., *op1Apply* or *op2Apply*, which can be seen in the respective output.

```
module m_solver
use m_operator
implicit none
contains
  ![Module procedures]
  subroutine solver(op,i,o)
  implicit none
    class(t_operator) :: op
    integer, dimension(1:), intent(in) :: i
    integer, dimension(1:), intent(out) :: o
    ![Some algorithm ...]
    call op%apply(i,o)
    ![...]
  end subroutine solver
end module m_solver
```

Listing A.5: solver.f90

```
program polymorphisms
use m_operator
use m_solver
implicit none

![Declarations]
integer, parameter :: ndim = 2
integer, dimension(ndim) :: i, o
type(t_op1), target :: op1
type(t_op2), target :: op2

![Set ndim in the operator structures]
op1%ndim = ndim
op2%ndim = ndim

![Initialize arrays]
i(:) = 1
o(:) = 0 ! values not used

![Call solver with operator op1 and input i]
call solver(op1,i,o)
write(*,*) o

![Call solver with operator op2 and input i]
call solver(op2,i,o)
write(*,*) o

end program polymorphisms

![Output]
!           1           1   (With t_op1: o = i)
!           2           2   (With t_op2: o = 2*i)
```

Listing A.6: polymorphism.f90

A.4. The ACE file format

Contents The ACE file format is a binary format that is used by BTDFT. An ACE file consists of two parts: A header with a fixed format, which contains data such as the total number of grid points and the number of orbitals, and a body, which contains, e.g., the density, orbitals, eigenvalues, etc. The format, content, and size of the body is solely determined by the header information. The exact contents of the ACE file format are described in the Doxygen documentation (see appendix A.5) and in the corresponding source code files in more detail.

Different ACE files can have different contents, depending on their source and purpose. E.g., an ACE file can contain orbitals or not and if the orbitals are stored, they can be real-valued or complex-valued. In any case, BTDFT can handle any valid ACE file as input with the only exception that an ACE file that stems from an initial guess or from a restart file of BTDFT_gs cannot be used as input to the propagator BTDFT_td since no orbitals are stored.

In particular, the following is possible:

- Create an ACE file from a special PARSEC output file using parsec2ace.
- Start a new ground-state calculation from any ACE file, even one that was output as final_state.ace by the propagator.
- Refine the accuracy of a previously calculated ground state or from a restart ACE file.
- Resume a propagation from a file that already contains a propagated state.

Binary format and portability The MPI programs BTDFT_gs and BTDFT_td read and write the ACE files using MPI_IO routines. These allow the MPI processes to open a file collectively and write or read in parallel, each MPI process its own data (implemented in the file BTDFT/src/io/io_interface.ace). The ACE file data are stored bytewise one after another in the memory, which requires the *native* data representation. This is explained in the MPI standard [MPI12] in detail.

In the sequential programs yACES_guess, ace2human, and parsec2ace the ACE files are accessed via Fortran's stream IO, which has been introduced with the Fortran 2003 standard [Ada+09]. The stream IO also allows to write (read) numerical values bytewise into (from) the memory. Thus, if one outputs a double precision value (8 Byte) after three integer values (4 Byte each), one knows exactly that the three integers occupy the bytes 0-3, 4-7, and 8-11 relative to the beginning of the file, while the double precision value occupies the bytes 12-17[50]. Since this is exactly the way the MPI_IO routines write the data to the memory, the communication between the parallel and sequential programs via ACE files works fine.

The native representation of data in the memory can differ between different computers. This restricts the portability of an ACE file. Hence, an ACE file that was generated at one computer may not be used at another computer since the representation of numerical values in the memory can be different[51].

[50] In record based Fortran IO, which is the default, this is not possible since the IO is organized in subunits called records.

[51] There is no portability problem between btrzx5 and btrzx3.

Other possibilities for the data representation of MPI_IO are *internal* and *external*32. *internal* guarantees portability but requires to use the same MPI implementation (e.g., Intel MPI, OpenMPI, or MPICH) to read the file at another machine. *external*32 is portable and can be read by any MPI implementation. However, the Fortran stream IO in the sequential BTDFT programs requires the ACE files to be stored bytewise in the file system's native representation (up to my understanding). Still, user defined data representations can in principle be used to restore portability [MPI12].

A.5. The Doxygen documentation

Doxygen [Hee16] is a program that creates a documentation (e.g., in html) by parsing and interpreting source code files directly. It was originally written for C but, by now, also supports Fortran code to some extent. In the source code files it recognizes, e.g., derived types and their components as well as module procedures or module data and creates links and dependency trees. Comments in the source code that are given in a specific format (including some keywords) are automatically added to the corresponding documentation. Additional documentation pages can be added and linked into the documentation.

Each of the five BTDFT programs has its own documentation with its own configuration file. The configuration files are located at BTDFT/build/dox/ and have the suffix '.cfg'. All options that can be controlled in the configuration file are explained in the manual. Among them are the project's name, a release tag, output and source directories, file filters, etc.

The second kind of file in BTDFT/build/dox/ has the suffix '.filter'. These files contain a bash script that preprocesses the source code files the same way it is done by 'make' during the compilation of the BTDFT code.

Each of the BTDFT programs has specific documentation contents located in the directory BTDFT/doc/[program]_dox/. The additional documentation files in the latter directories must be written as C comments.

The documentation can be compiled in BTDFT/build/ by calling 'make doc'. Its output is generated in BTDFT/doc/doxygen/[program]/html/ and can be opened in any browser (open the index.html in the former directory). The main documentation is the one of the propagator BTDFT_td. On the main page of each of the program documentations, there is some explanation about how to run the program and, in the case of ace2human and parsec2ace, a list of command line arguments. For the three main programs there are some subpages that contain, e.g., the layout of the ACE file, the release history, explain the configuration parameters, or describe how to implement typical things like new parameters or external potentials. Moreover, all source code files are listed in the documentation with their contents as given by the comments inside the source code files.

A.6. Version control with Git

How version control with Git works is best explained in some tutorials or books available in the web [Git; CS14]. In the following, I introduce a few of the most important features and commands to use Git in the context of BTDFT.

Currently, the root Git repository is located in the file system of the IT service center of the UBT. The root repository is a pure repository without working directory to allow for 'push' commands as explained below. There is one master branch and one extra branch for each person who works with the code to allow for a parallel development.

There are different user privileges: normal users and administrators. The only difference between them is that only administrators are allowed to 'push' commits onto any branch of the root repository, specifically the master branch. Normal users can only 'push' onto their own branch named by their own user identifier (usually the bt- or s1- identifier) to prevent confusion. If a piece of code shall enter the master branch of the program, I recommend to merge the branch of the normal user into the local master branch and let the administrator do the 'push'. This behaviour is controlled by Git hooks and explained in one of the following paragraphs.

Getting started First of all, practical information on how to use git in the context of BTDFT can also be found in the file BTDFT.README, which is located in the same directory as the root repository itself.

To start with an own working copy of BTDFT, the root repository must be cloned. This is done by 'git clone [parent repository]' in the directory where the local working repository shall be generated. This way, a new directory called BTDFT/ will appear with the underlying file tree (see appendix A.1).

Change to the BTDFT/ directory ('cd BTDFT/'). By default, only the master branch is checked out. A list of existing branches can be printed using 'git branch'.

To set up a new branch with the bt-identifier, call 'git branch [identifier]'. To check out the new branch, call 'git checkout [identifier]'. In the local repository, there should now be a new branch labeled with the identifier (check with 'git branch'). To generate the branch with the same identifier in the remote root repository, type 'git push origin [identifier]'. The 'origin' keyword in the last command identifies the remote git repository the local repository was cloned from. As a normal user, this does only work with your own bt-identifier. Still, every one can pull every other branch from the remote root directory by 'git pull [remote repository identifier] [branch name]'.

Useful commands

- Add a (modified) file to the git repository (or more precisely: to the staging area): 'git add [file]'
- Output status information about files that are untracked, modified, etc.: 'git status'
- Ignore files that fit a certain pattern: Edit the file BTDFT/.gitignore
- Commit your changes (after you added them via 'git add' to the staging area): 'git commit'
- 'git commit' asks you to comment the commit. The commit history (log) is output via 'git log'
- List tags: 'git tag'

- Show the difference between the current working directory and the last commit: 'git diff'
- 'git diff' can also be called with further arguments to list the differences between arbitrary commits.
- Create a new branch: 'git branch [branch name]'
- Switch to a branch (checkout): 'git checkout [branch name]'
- Merge one branch ('feature') into another branch ('develop'):
  - 'git checkout develop'
  - 'git merge feature'
  - If necessary: resolve merge conflicts manually and commit your changes
- Delete a branch: 'git branch -d [branch]'
- Restore a file to the state of your last commit: 'git checkout -- [file]'
- Create a tag: 'git tag -a [tag name] -m [comment]'

Workflow In the work with BTDFT, Git can be ignored completely. Nevertheless, it helps a lot with the development. This is especially true if there is more than one programmer since the coordination of different code versions and merges is quite simple. Still, even for a single developer, it is quite handy to go back and forth in the commit and release history to track changes and bugs. Therefore, I present the workflow I used in the development of BTDFT in the following (see [Git; CS14]).

In the local repository, there should be at least two branches: *master* and the branch of the user labeled with the bt-identifier, e.g., *bt*123456. The master branch contains the official version of BTDFT and can be updated from the remote repository *origin* via 'git pull origin master' if there has been a new release. The branch *bt*123456 should be the own stable branch.

For the development, a local development branch *develop* can be generated through 'git branch develop' and 'git checkout develop'. After the implementation of new features, the changes can be merged back into the stable branch via 'git checkout bt123456' and 'git merge develop'. The *develop* branch itself can also be branched into several feature branches. Branching in Git is very cheap, since it only uses pointers to certain commits. Thus, extensive branching is recommended. Also, tagging important commits is recommended since they are easily checked out later.

Publishing via 'push' In order to publish the new code, the own bt-branch (e.g., *bt*123456) can be pushed onto the corresponding branch of the remote repository. This is done by 'git checkout bt123456' and 'git push origin bt123456'. To push tags, run 'git push --tags'. If the work shall enter the master branch the project administrator can merge the *bt*123456 branch into the master branch and push it.

Project administrator As mentioned before, only the project administrator can 'push' onto the root master branch while normal users are only allowed to push onto their own branch in the root repository. The corresponding settings are done via Git hooks in the file 'hooks/pre-receive' inside the root repository.

Generation of the root repository The information in this paragraph explains, how the root repository has been setup and is just for documentation. The root repository has been created from an initially existing git repository which is called *local* in the following.

- Change directory to the location of the new root repository
- Make directory for the repository:
  'mkdir BTDFT.git'
  'cd BTDFT.git'
- Initialize git with options '--bare' (no working directory, 'push' is only allowed to a bare repository) and '--shared' (repository is shared, all who have write access are in principle allowed to push):
  'git init --bare --shared'
- Change the group of the repository to 'btpc', allow group write access and restore the SGID bit for all directories (applied to directories, this ensures that the access rights of a directory are handed down to the child directories and files. The SGID bit is set by the '--shared' option of 'git init' but destroyed by the 'chgrp' command below. Therefore, it has to be restored because otherwise no one would be allowed to 'push' to the remote repository):
  'chgrp -R btpc BTDFT.git'
  'chmod -R g+w BTDFT.git'
  'chmod -R g+s 'find BTDFT.git -type d''
- Add the new remote repository as *origin* to the local repository and 'push' the branch with the latest BTDFT version to the new remote repository:
  'cd [local repository]'
  'git checkout master'
  'git remote move origin origin_old' (move the old *origin* repository to *origin_old* or remove it if it is not used anymore via 'git remote remove origin')
  'git remote add origin [remote repository]' (add the new remote repository as new *origin*)
  'git push origin master' ('push' the local *master* branch onto the new remote *master* branch)

B. Computer clusters in Bayreuth

The main specifications of the UBT clusters are listed below. The corresponding nominal[52] data can be found in the web [CPUc; CPUd; CPUb; CPUa; Intb; Intc; Inta] and the web pages of the UBT IT service center (for members of the UBT). Schematics of the processors that are built into the UBT nodes are shown in Figure B.1.

B.1. btrzx5

nodes001–node201

Node	2 sockets
CPU	2× Intel(R) Xeon(R) E5520 (Intel Core Bloomfield)
Total Cores	2×4
Clock rate	2.27 GHz
Memory	2×12 GB
Peak (node)	72.64 GFlop/s
SIMD	SSE
Bandwidth	2×25.6 GB/s
L1D	$2 \times 4 \times 32$ kB, each shared among 1 thread
L2	$2 \times 4 \times 256$ kB, each shared among 1 thread
L3	$2 \times 1 \times 8$ MB, each shared among 4 threads

nodes205–node256

Node	2 sockets
CPU	2× Intel(R) Xeon(R) E5620 (Intel Core Westmere)
Total Cores	2×4
Clock rate	2.40 GHz
Memory	2×12 GB
Peak (node)	76.80 GFlop/s
SIMD	SSE
Bandwidth	2×25.6 GB/s
L1D	$2 \times 4 \times 32$ kB, each shared among 1 thread
L2	$2 \times 4 \times 256$ kB, each shared among 1 thread
L3	$2 \times 1 \times 12$ MB, each shared among 4 threads

nodes257–node264

Node	2 sockets
CPU	2× Intel(R) Xeon(R) E5-2670 (Intel Core SandyBridge EP)
Total Cores	2×8
Clock rate	2.60 GHz
Memory	2×32 GB
Peak (node)	332.80 GFlop/s

[52] The nominal bandwidth usually deviates usually from a measured one. In practice one cannot expect to get higher bandwidths than those measured with the STREAM [McC95] benchmarks.

SIMD	AVX
Bandwidth	2×51.2 GB/s
L1D	$2 \times 8 \times 32$ kB, each shared among 1 thread
L2	$2 \times 8 \times 256$ kB, each shared among 1 thread
L3	$2 \times 1 \times 20$ MB, each shared among 8 threads

B.2. btrzx3

2-socket nodes

Node	2 sockets
CPU	2× AMD Opteron(tm) 6348 (AMD Interlagos)
Total Cores	2×12 (Each two build a Piledriver module and share one FPU)
Clock rate	2.80 GHz
Memory	4×16 GB
Peak (node)	268.80 GFlop/s
SIMD	AVX
Bandwidth	2×51.2 GB/s
L1D	$2 \times 12 \times 16$ kB, each shared among 1 thread
L2	$2 \times 6 \times 2$ MB, each shared among 2 threads
L3	$2 \times 2 \times 6$ MB, each shared among 6 threads

4-socket nodes

Nodes: r05n37, r05n38, r12n33, r12n34, r14n01, r14n02, r14n03, r14n04

Node	4 sockets
CPU	4× AMD Opteron(tm) 6348 (AMD Interlagos)
Total Cores	4×12 (Each two build a Piledriver module and share one FPU)
Clock rate	2.80 GHz
Memory	8×32 GB
Peak (node)	537.6 GFlop/s
SIMD	AVX
Bandwidth	4×51.2 GB/s
L1D	$4 \times 12 \times 16$ kB, each shared among 1 thread
L2	$4 \times 6 \times 2$ MB, each shared among 2 threads
L3	$4 \times 2 \times 6$ MB, each shared among 6 threads

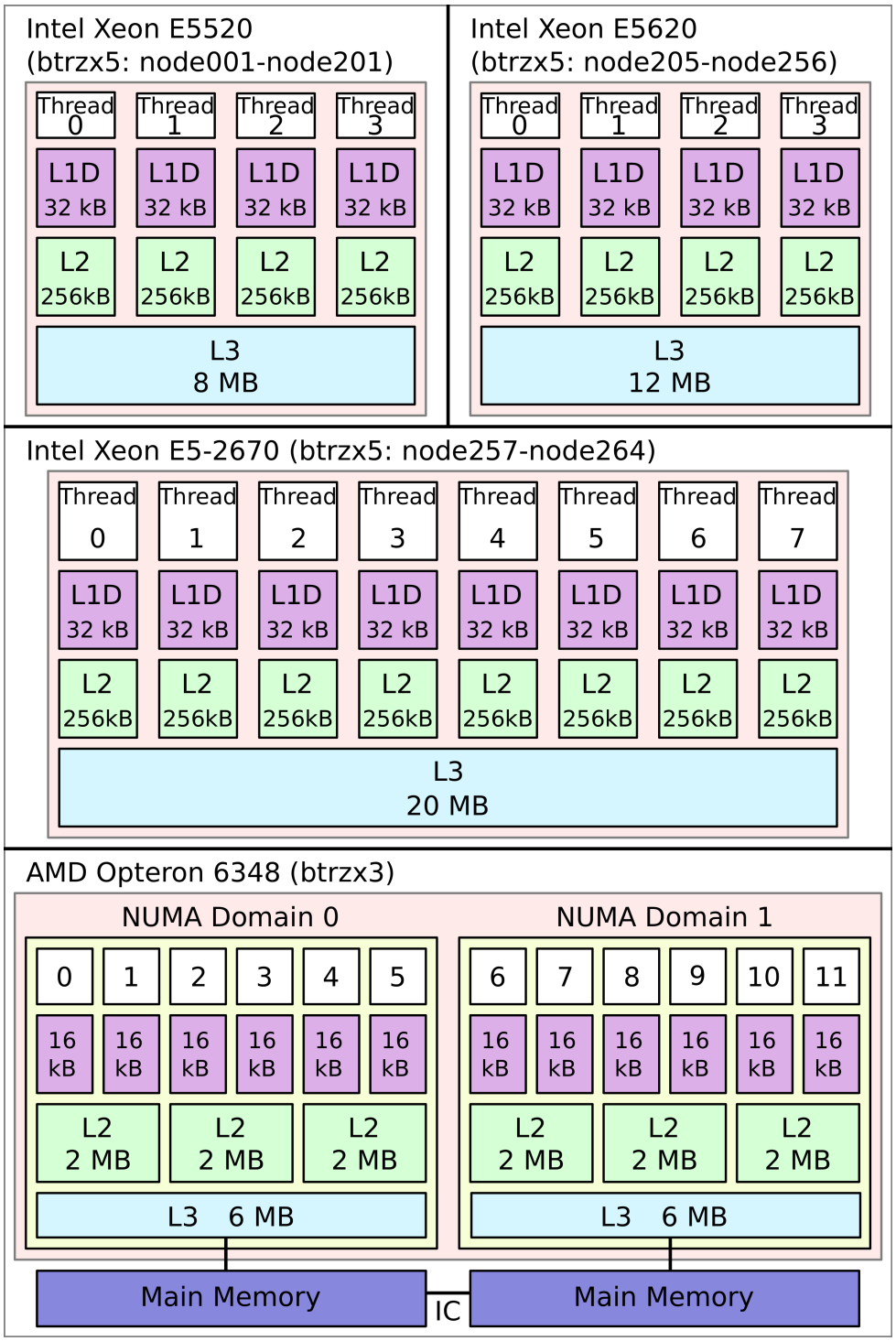


Figure B.1: Processors that are built onto the sockets of the UBT nodes.

C. Additional benchmarks and performance engineering

Since this work shall also serve as an additional documentation of BTDFT, I discuss several performance aspects in this section. However, this summary is far from being complete or very detailed but just an outline of techniques and thoughts that I found useful in the course of writing BTDFT. Most of the examples and explanations are takes from [HW10], supported by my own measurements, and applied in the context of BTDFT.

C.1. Latency and bandwidth in parallel networks

The communication between two MPI processes in a network can be described by two measures. These are the latency T_l, which is the time it takes for sending a message of zero size, and the bandwidth B, which is the maximum rate at which data can be sent across the network[53]. Both are in principle determined by the network technology, the transmission protocols used, and the distance of the communication partners within the network topology.

Both, latency and bandwidth, can be combined into an effective bandwidth that measures the effective data transfer rate depending on the size of the message N_{message}. It reads [HW10]

$$B_{\text{eff}} = \frac{N_{\text{message}}}{T_l + \frac{N_{\text{message}}}{B}} \tag{C.1}$$

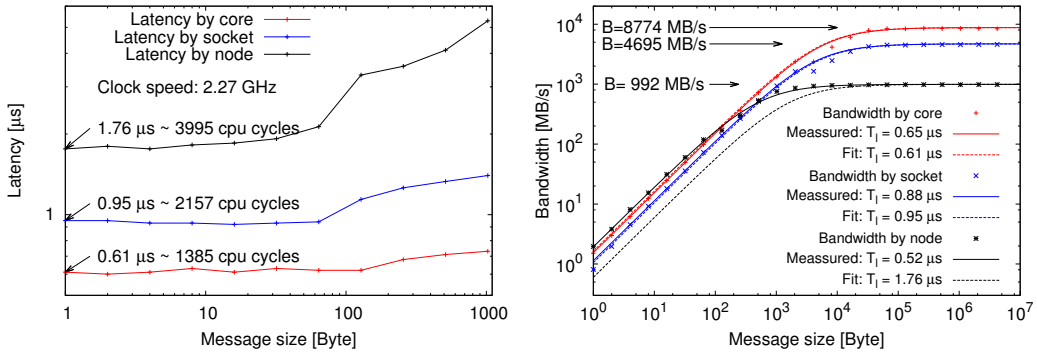
which is essentially the size of the message divided by the total time it takes for sending it. From this definition one can see that for small message sizes the effective bandwidth is capped by the latency whereas for large message sizes it converges to the theoretical bandwidth B. Therefore, the range at which the communication is latency or bandwidth dominated depends on their ratio.

Latency and bandwidth can be measured, e.g., by using the OMB suite [Pan16]. The results are shown in figure C.1 for point-to-point communications between MPI processes sitting on two different nodes (i, by node) and on the same node, once in different NUMA domains (ii, by socket) and once in the same NUMA domain (iii, by core). Thus, the first measurement (i) tests the network between the two nodes (170 and 171 on btrzx5), the second measurement (ii) the IC between the NUMA domains (in this case the processors) on node 171, and the third measurement (iii) essentially a memory-to-memory copy without network in between.

The latency measurement is done for small messages with the latency being the limit for $N_{\text{message}} \to 0$. The values shake a little but one can clearly see that, as expected, the latency is the larger the slower the network connection is. For the 2.27 GHz clock frequency of the underlying processor, the inter-node latency takes about 4000 processor cycles.

The bandwidth measurement in figure C.1b is done for various message sizes. Hence, one should effectively measures B_{eff} from equation (C.1). As expected, the effective bandwidth for large messages converges, e.g., for the inter-node communication towards 1 GB/s (which also is the nominal bandwidth of the network between these

[53] Note that the bandwidth from a point-to-point communication with one node sending and one receiving is usually different from the bandwidth measured if both nodes send and receive data (bidirectional bandwidth) or if the communication is collective.



(a) Blocking latency measurement. (b) Non-blocking bandwidth measurement.

Figure C.1: Latency and bandwidth measurements with the OMB benchmark suit [Pan16] on the nodes 171-170 of btrzx5 between the nodes (i,by node), between the processors (NUMA domains) on one node (ii, by socket) and within a single processor (iii, by core).

nodes). Compared to that the inter-processor bandwidth (ii, by socket) on the same node is more than four times and the intra-processor bandwidth (iii, by core) is more than eight times larger. For smaller message sizes the latency becomes important and hampers the communication. The dashed lines in the figure show the effective bandwidth according to equation (C.1) with the latency taken from the latency measurement and the bandwidth fitted for large message sizes.

The solid lines also show the effective bandwidth but with the latency fitted to the small-message region. For the two intra-node measurements (ii) and (iii) both fit well to the data within the uncertainties of the latency measurement. However, the latency fitted to the bandwidth data of the inter-node (i) communication is much smaller than the one measured before in the latency measurement. Up to my understanding, this is because the latency measurement uses blocking MPI communication while the bandwidth measurement uses non-blocking MPI communication [MPI12; Pan16], which hides a part of the latency.

Finally, the kink that appears in the transition region between the latency and bandwidth dominated parts of the bandwidth measurements can not be described by means of the effective bandwidth from equation (C.1). This indicates that the underlying implementation and technology is more complicated. [HW10]

As a conclusion, one can state that non-blocking MPI communication can, if supported by the architecture, hide latency times [MPI12]. Moreover, sending one larger message instead of many small message has clear advantages since the latency only has to be paid once[54]. The latter does not apply if one of the 'small' messages is already large enough to be in the bandwidth dominated part of the effective bandwidth. Examples are the halo layers of the KS orbitals. In this case, doing a collective halo communication in a single message for all orbitals at the same time typically gives no improvement.

[54]For data that are non-contiguous in the memory, derived MPI types can be used [MPI12].

C.2. Node-level hardware parallelization

A single floating point operation (Flop), i.e., MULT or ADD, takes 4-5 processor cycles to be processed. However, a single core in a modern processor can perform up to (theoretically) 4 Flops per processor cycle, which is because of the hardware parallelization in the cores I briefly introduce this in the following according to [HW10].

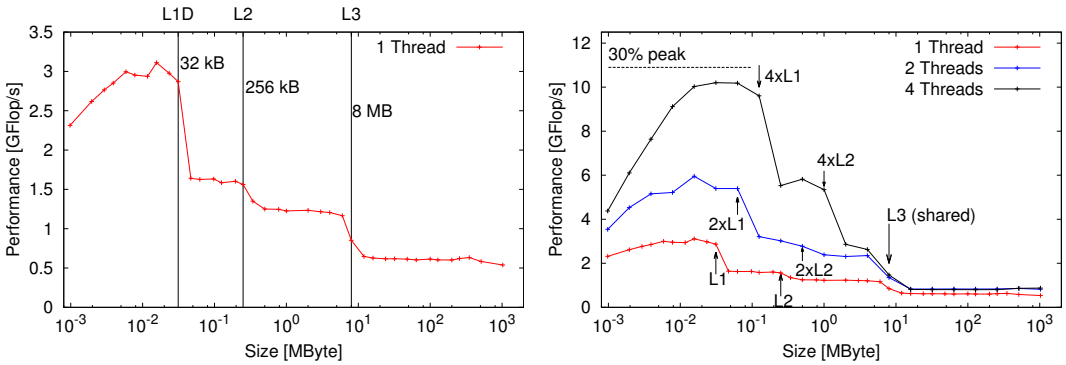
The theoretical number of up to 4 Flops per cycles results in the so-called peak performance, which is an often stressed measure for the performance of computers. Still, it is a quite theoretical value one can only come close to with highly optimized benchmark codes. For example, in the benchmark test in appendix C.3 I reached a maximum of 30% of the nominal peak performance on a UBT node.

Pipelining Pipelining works as assembly-line work in industry. In order to perform a Flop like an ADD operation, the data have to pass several stages whereas each stage takes a processor cycle. The first stage is, e.g., to fetch the data. When the first DP word passed the first stage and moves on to the second stage, the second DP word can already enter the first stage. At the moment the pipeline is completely filled the core can output one result per cycle, which is a huge performance boost compared to the 4-5 cycles a single stand-alone operation takes.

SMT The pipelining does usually not work perfectly, which leads to gaps in the pipeline called pipeline bubbles. Simultaneous multithreading (SMT) is used to run a second thread per core. Both threads share most of the resources such as the cache and, necessarily, the floating point unit (FPU) but can together eventually make better use of the pipeline. However, if a computation is sensitive to the cache size, this hampers the performance since the caches are shared between the SMT threads. Intel's version of SMT is Hyper-Threading whereas AMD's approach uses hardware threads that form Bulldozer or Piledriver modules. The latter consist of two physical cores that share resources such as the FPU (see appendix B.2).

SIMD vectorization The idea of SIMD ('single instruction multiple data') vectorization is to apply an operation not to a single DP word but on a vector of DP words at a time. If a processor supports SIMD vectorization can be seen from the instruction set it supports. The SSE instruction set works on SIMD registers with a width of 2 DP words (e.g., most btrzx5 nodes), the AVX instruction set works on SIMD registers with a width of 4 DP words (e.g., btrzx3) (see appendix B). While the pipelining improves the theoretical performance to 1 Flops/cycle, the additional SIMD vectorization raises this value to 2 (SSE) or 4 (AVX) Flops/cycle.

Peak performance The peak performance of a processor or node in Flops per second can then be calculated from the number of FPUs (per processor or node) times 2 (ADD and MULT operations count separately since they can operate in parallel) times the number of Flops per cycle per FPU times the number of cycles per second, which is the clock frequency of a processor. This results in the values for the peak performance given in appendix B. Note that each two of the cores on the btrzx3 nodes build a Piledriver module and thus share a single FPU.



(a) Performance for one thread.

(b) Performance for up to four threads.

Figure C.2: Performance of the vector triad on one socket of the btrzx5 node193 (Intel Xeon E5520, four cores) for different sizes of the working set (logarithmic scale). Performance drops appear when the size of the working set exceeds the (combined) size of a cache level. For larger working sets the performance is capped by the memory bandwidth. At best about 30% of the nominal peak performance is reached.

Conclusion In view of the main memory latency of about 100 cycles and a cache-line size of typically 8 or 16 DP words[55] the high theoretical performance of a processor underlines the necessity to think about efficient memory access and cache usage. The effects of the hardware parallelization from above can only be used if the data are loaded fast enough.

C.3. Impact of the cache hierarchy

The vector triad $A(:) = B(:)+C(:)*D(:)$ [HW10], where A, B, C, and D are DP arrays of dimension $ndim$, is an often used benchmark kernel. MULT and ADD operations, which can in principle be done in parallel by the FPU, are balanced in the vector triad. For the computation the arrays B, C, and D have to be loaded and the array A must be stored (+ 1 write allocate). Hence, there are in total four (five) steams[56] that have to pass the memory interface. Therefore, the computation of one component of A requires two Flops and $4(5) \times 8\,\text{Byte} = 32(40)\,\text{Byte}$.

The following code serves for the performance measurement of the vector triad [HW10] The conditional in line 5 prevents the compiler from interchanging the two loops. It therefore enforces that the arrays are loaded in each iteration.

Figure C.2 shows the performance of the vector triad on a single socket of the btrzx5 node193 (Intel Xeon E5520, see figure B.1 top left) depending on the total size of the arrays A, B, C, and D in the memory. In figure C.2a the performance of a single thread (i.e., without parallelization) increases first and then shows sharp drops each time the size of the working set exceeds the size of a cache level. The drops can be explained by the lower bandwidths and higher latencies of upper level caches. Therefore, if the data are too large to fit into a certain cache level, they have to be loaded from the next

[55] I.e., the main memory latency has to be paid once for each 8 or 16 DP words.
[56] 3 LOAD + 1 STORE (+ 1 write allocate).

```
do j = 1, niter
 do i = 1, ndim
   A(i) = B(i) + C(i)*D(i)
 end do
 if ([something that is never true]) call dummy(A,B,C,D)
end do
```

Listing C.1: Vector triad [HW10].

higher cache or the main memory, which takes more time. The performance for large working sets is finally capped by the main memory performance. The above mentioned hardware parallelization techniques such as pipelining and SIMD vectorization work inefficient if the array sizes are very small, which explains the initial performance increase. [HW10]

If the inner loop is parallelized with OpenMP (figure C.2b), the performance increases in the non-memory-bound region and reaches up to about 30% of the peak performance of the processor. Since the working set is equally distributed among the threads, the total L1D and L2 capacities get larger proportional to the number of threads, i.e., the performance drops appear at larger working sets. The L3 drop, however, always appears at the same size since the L3 cache is shared among all threads (see figure B.1 top left) and already fully available for a single thread. The memory-bound performance at the large-size end of figure C.2b already saturates for two threads. This is because the total memory bandwidth is as well shared between all threads on the processor.

This example demonstrates the effect of the different cache levels and of sharing resources among threads (L3 cache and memory bandwidth). The BTDFT data sets usually exceed the typical cache sizes by far and BTDFT operates in the memory-bound region and must handle these shared resources. This demonstrates that it makes sense to use data that still reside in the cache as often as possible and do more computations per byte loaded from the memory. This is the subject of appendix C.4.

C.4. Code optimization made simple

C.4.1. General approach

In this section I briefly introduce some simple rules to improve the memory access behaviour of typical code samples. The general rules follow [HW10], which I recommend as reference.

When optimizing a program as BTDFT that consists of many thousands of lines of source code, one must make two decisions: First, which part of the code shall be optimized and, second, how can this code be optimized. In other words, is the code efficient enough to hit a bottleneck such as the memory bandwidth saturation and how can the code be optimized in this respect.

In BTDFT, as in most scientific codes [HW10], the memory bandwidth is the relevant bottleneck since one operation is usually applied to a large set of data that does not fit into any of the cache levels and has to be loaded each time a new operation is applied. Therefore, I mostly discuss this topic, i.e., how to do more useful computations per byte loaded from the memory.

In any case, most of the optimization is finally done by the compiler as shown in figure 3.7 and again as the left part of figure C.3. It is therefore mandatory to help the compiler to optimize the code and not to hide performance critical operations behind subroutines or index arrays as discussed in section 3.4.

C.4.2. Easy rules

Expensive operations and subexpressions The evaluation of a sine, an exponential function, or a power function costs a lot of processor cycles, which partly can be avoided. The square of a number, e.g., can be computed as a^2 or $a \cdot a$, where the latter is much cheaper. This is especially true if the operation is used in a subexpression in some loop as in listing C.2. In this case, $s + r * exp(x)$ can as well be calculated once before the loop and stored in a temporal variable[57]. [HW10]

```
1 do i = 1, n
2  A(i) = A(i) + s+r*exp(x)
3 end do
```

Listing C.2: Avoid subexpressions.

```
1 do j = 1, nj
2  do i = 1, ni
3   if (i>j) then
4    [...]
5   else
6    [...]
7   end if
8  end do
9 end do
```

Listing C.3: Branches in inner loops.

Branches in loops Branches in the innermost of a nested loop as in listing C.3 should be avoided. In order to keep the data flow at a constant rate, the processor tries to guess the outcome of a condition, which is called branch prediction [HW10]. If the guess is wrong but the related data are already in the pipeline, the pipeline is flushed completely. To avoid this the nested loop in listing C.3 can be divided into two nested loops, one for $i > j$ and one for $i \leq j$. [HW10]

Copy operations As discussed in section 3.4.1, every copy operation requires data to be loaded from the memory, especially if the copied data are large. In many cases copy operations can be avoided. One example is given in listing C.4 in which the arrays a and b are given to some subroutine and change their roles at the end of the loop. There, a is copied to a temporary array tmp, b is copied to a, and finally tmp to b, i.e., in total are the contents of a and b switched.

Copying of these data can be avoided: First, by using a two-dimensional array $a(:,:)$ with $a(:,1)$ being the former a and $a(:,2)$ being the former b [HW10]. Instead of switching the data, it is sufficient to switch the second index (listing C.5). Second, by using pointers pa and pb that are initially assigned to a and b. In this case, only the pointer's references are switched at the end of the loop (listing C.6).

[57] This sounds simple. However, the strict division of BTDFT in an initialization part and a calculation part as introduced in appendix A.3.1 is the same, just on another scale.

```
1 !
2 !
3 do i = 1, n
4   call dummy(a,b)
5   tmp(:) = a(:)
6   a(:) = b(:)
7   b(:) = tmp(:)
8 end do
```

Listing C.4: Copy 1

```
1 i1 = 1
2 i2 = 2
3 do i = 1, n
4   call dummy(a(:,i1),a(:,i2))
5   itmp = i1
6   i1 = i2
7   i2 = itmp
8 end do
```

Listing C.5: Copy 2

```
1 pa => a(:)
2 pb => b(:)
3 do i = 1, n
4   call dummy(pa,pb)
5   ptmp => pa
6   pa => pb
7   pb => ptmp
8 end do
```

Listing C.6: Copy 3

Loop fusion A further example to improve memory access is shown in listing C.7. There, the arrays a and b are summed up into res and the norm of res is computed. In this example a and b are loaded in line 3 and res is stored (+ 1 write allocate). In line 4 res is again loaded to calculate the norm, which results in 3 LOAD streams and 1 STORE stream (+ 1 write allocate).

Lines 3 and 4 can be seen as two loops. res is used in both lines but is too large to stay in the cache in between and is loaded twice. This can be resolved by loop fusion [HW10]. One can combine both loops and reuse each element of res for the norm as soon as it has been calculated as shown in listing C.8. This way, one saves one LOAD stream of the res array and therefore about 20%[58] of the computation time. [HW10]

Smart compilers might be able to optimize the given code by themselves. Still, if the situation is more complex, e.g., if calculating the norm is initially outsourced into an extra subroutine, loop fusion might work fine. I could, e.g., increase the performance of my naively written CG solvers by about 30% only by applying loop fusion.

```
1 !
2 !
3 res(:) = a(:) + b(:)
4 norm = sum(res(:)*res(:))
5 !
6 norm = sqrt(norm)
```

Listing C.7: Loop fusion 1

```
1 norm = 0.0
2 do i = 1, n
3   res(i) = a(i) + b(i)
4   norm = norm + res(i)*res(i)
5 end do
6 norm = sqrt(norm)
```

Listing C.8: Loop fusion 2

An example that has a similar scope is the application of an operator such as $1+a\cdot\hat{H}$ as it appears in the Crank-Nicolson propagator. In a code as BTDFT there usually exists a subroutine that applies the Hamiltonian $\hat{H}$. Hence, the intuitive way to apply $1+a\cdot\hat{H}$ to an array psi is to apply $\hat{H}$ to psi and store the result in $hpsi$ ($hpsi(:) = H*psi(:)$) a first step, and add those in a second step to $hpsi(:) = psi(:) + a*hpsi(:)$. For this, one in total requires to load psi twice, load $hpsi$ once, and store hpsi twice (+1 write allocate for the first line of the code[59]). Including write allocate, this results in six streams (not counting $\hat{H}$).

This code can be improved significantly by implementing a new operator that directly applies $\hat{H}_{\text{new}} = 1 + a\cdot\hat{H}$. If the relevant performance bottleneck is the memory bandwidth, its application requires as much computation time as the application of

[58] If LOAD, STORE, and write allocate are counted equally.

[59] The second step does not require a write allocate since $hpsi$ is in any case loaded for the right-hand side of the source line.

$\hat{H}$ but saves the second step from above. This way, *psi* is loaded only once and *hpsi* is stored once (+ 1 write allocate), i.e., three streams including write allocate. This reduces the computation time by about 50% (making the same assumptions as above).

Strided memory access The memory access as well as the communication between the cache levels is organized in cache lines of usually 8 or 16 DP words to hide latency times. If the memory access to some large array is strided, e.g., only every second value is used, a significant part of the bandwidth is wasted.

```
1 do j=1,n1
2   do i=1,n2
3     a(i,j) = [...]
4   end do
5 end do
```

Listing C.9: Nested loop over a 2D array

For the same reason the access of a two-dimensional array in a nested loop is sensitive to the order of the loops. Fortran stores multi-dimensional arrays column-wise, i.e., the first index is the one that is contiguous in memory. A nested loop should therefore run through the array as in listing C.9. If the order of the two loops is interchanged and the second dimension of the array is large enough that a previously loaded cache line is evicted from the cache before the next time it is needed, $7/8^{\text{th}}$ (for a cache-line size of 8 DP words) of the theoretical bandwidth is wasted and the latency time has to be paid for each single element.

A more involved example of strided memory access, in which one must find a compromise between different strides, is the calculation of the density from the KS orbitals. This is similar to the code in listings C.10 and C.11 with a complex-valued orbital array *orb* and a real-valued density array *dens*. In listing C.10 the density is summed up orbital-wise, i.e., the outer loop runs through the orbitals whereas the inner loop runs through the grid points. This case seems to be the natural choice since the orbital array *orb* is always accessed continuously. However, since the single orbitals are large, the density array *dens* must be loaded and stored anew for each orbital.

In listing C.11 the density is summed up grid-point-wise and the situation is reversed. The density array is loaded only once but how often the orbital array is loaded depends on the total number of orbitals *norb*. If *norb* is small, i.e., at least one cache line of each orbital fits into the relevant cache, the orbital array is also loaded once only. However, if *norb* gets larger, orbital cache lines that are needed in the near future might already be evicted and must be loaded again from the next higher cache level. Since the computation itself is very primitive, a low level cache such as the L1D might already be the relevant one in this case. This means that *norb* must be quite small for this code to work well.

```
do iorb = 1, norb
 do i = 1, ndim
  dens(i) = dens(i) + conjg(orb(i,iorb))*orb(i,iorb)
 end do
end do
```

Listing C.10: Density 1

```
do i = 1, ndim
 do iorb = 1, norb
  dens(i) = dens(i) + conjg(orb(i,iorb))*orb(i,iorb)
 end do
end do
```

Listing C.11: Density 2

```
do lb = 1, norb, stride
 do i = 1, ndim
  ub = lb + stride - 1
  do iorb = lb, ub
    dens(i) = dens(i) + conjg(orb(i,iorb))*orb(i,iorb)
  end do
 end do
end do
```

Listing C.12: Density 3

These two cases are only boundary cases of a more flexible approach in listing C.12. There, the total number of orbitals is divided into blocks with a fixed number of orbitals[60] (*stride*). *lb* is the lower bound and *ub* the upper bound of each block. Inside each block, the number of orbitals is assumed to be small and the code from listing C.11 is efficient. The density array must be loaded once per block. $stride = 1$ results in the code of listing C.10 whereas $stride = norb$ results in the code of listing C.11. In some tests, I found a global performance maximum at strides between four and seven on one of the local workstation computers and around 16 on the btrzx5 node260 (Intel Xeon E5-2670). This example is related to the efficient application of dense matrices as discussed in [HW10].

C.4.3. Remarks about OpenMP and the 3D Jacobi smoother

Listing 3.1 on page 30 shows the code of the 3D Jacobi smoother, which I used for the introduction of the memory bandwidth bottleneck in section 3.4.1. As in the vector triad (listing C.1) the outermost loop simply repeats the Jacobi smoother to get macroscopic computation times for the performance evaluation. The dummy subroutine again fools the compiler and prevents it from doing unwanted optimizations.

The Jacobi smoother inside this loop consists of three nested loops that run through the three dimensional cubic grid with dimensions ni, nj, and nk. x is the real-valued function (as a three-dimensional array) that is smoothed and y is the result. b is some

[60] If the total number of orbitals is not divisible by the stride, the remaining orbitals must be handled in a smaller block.

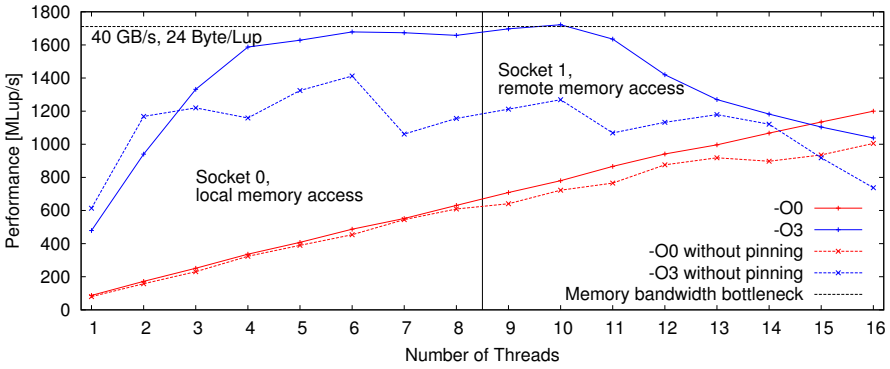


Figure C.3: Parallel performance of the Jacobi smoother [HW10] on both sockets of the btrzx5 node258 (2× Intel Xeon E5-2670) with and without compiler optimization and thread-pinning.

real value.

The outermost of the nested loops is parallelized with OpenMP, which is done through the comment-like lines in the listing. These tell the program where a parallel region begins and ends and how the parallelization shall be done. In this case, the third dimension nk is equally distributed among the OpenMP threads.

The parallelization with OpenMP looks easy at the first glance but it is easy as well to write code with a poor efficiency. As an example, I again show the performance of the Jacobi smoother [HW10] from section 3.4.1 with and without compiler optimization but with the parallelization extended to the second socket of the node (see figure C.3). In the naive implementation I used, there is one process with one master thread pinned to socket 0. In a parallel region of the OpenMP code the master thread opens a number of additional threads that are first pinned onto socket 0 until it is fully occupied. The remaining threads are pinned to cores on socket 1.

As already discussed, the memory bandwidth saturates if the threads are only pinned onto socket 0. One would naively expect a performance increase when additional threads are pinned to socket 1 since they could utilize an independent memory interface. However, the optimized (blue) curve in figure C.3 does not exceed the memory bandwidth bottleneck of the single socket for a parallelization beyond socket 0 and the performance finally decreases again. This is because in the implementation shown in listing 3.1 only the master thread on socket 0 allocates arrays in its local memory, i.e., the memory attached to socket 0. Threads pinned to socket 1 use a remote memory access across the intra-node IC network, which hampers the performance.

In order to use OpenMP parallelization efficiently, e.g., in addition to MPI, one possibility is to run one MPI process per NUMA domain (in this case one MPI process per socket). Each MPI process opens up to eight threads that are pinned to the domain of the respective process. Defining such thread domains in combination with MPI processes can be done by either using LIKWID [THW10] to start the job or via Intel's MPI implementation (other MPI implementations should support this as well).

By default, the threads are not pinned to a specific core or domain (so far, I always pinned OpenMP threads as far as used). In this case, the threads are moved around within the ccNUMA node by the operating system. This causes, among others, casual

remote memory access and related performance drops. [HW10]

The related performance without thread-pinning is shown as dashed lines in figure C.3. One clearly sees that the performance without pinning is usually lower than the one with pinning (the case discussed so far)[61] and that there is no clear performance trend depending on the number of threads. A more detailed discussion can again be found in [HW10].

[61] The higher performance of the not-pinned run with -O3 optimization for one and two threads could be due to the different compilers I used for calculations with and without pinning.

D. Proof concerning density fluctuations in a donor-acceptor model

In this section, I present the proof that

$$\underbrace{\left[\cos(\tilde{\omega}t)\sin(\bar{\omega}t) - a\sin(\tilde{\omega}t)\cos(\bar{\omega}t)\right]^2}_{f(t)} = \underbrace{\left[1-(1-a^2)\sin^2(\tilde{\omega}t)\right]}_{g(t)} h(t) \tag{D.1}$$

with the envelope

$$g(t) = 1-(1-a^2)\sin^2(\tilde{\omega}t) = \cos^2(\tilde{\omega}t) + a^2\sin(\tilde{\omega}t) \tag{D.2}$$

and a fast oscillating function $h(t)$ with time-dependent frequency but an amplitude of one without beat.

In comparison to section 5.4.1, $a = \frac{\Delta E}{\sqrt{\Delta E^2+V^2}}$ with $-1 \leq a \leq 1$ measures the ratio between the energetic off-resonance ΔE between the donor and acceptor excited states and their coupling strength V. The limiting cases $a = \pm 1$ mean 'no coupling' and $a = 0$ 'resonant coupling'. The angular frequencies that appear in equation D.1 are defined by $\hbar\tilde{\omega} = \sqrt{\Delta E^2 + V^2}$ and $\hbar\bar{\omega} = \bar{E} - E_0$ with typically $\bar{\omega} \gg \tilde{\omega}$. In the following, I will assume $0 \leq a \leq 1$, i.e., the acceptor excited state has equal or higher energy than the donor excited state. The proof for $a < 0$ works essentially the same. Below, I make comments where something changes.

Rewrite $f(t)$

$f(t)$ can be decomposed into a Fourier series.

$$\begin{aligned} f(t) =& \left[\cos(\tilde{\omega}t)\sin(\bar{\omega}t) - a\sin(\tilde{\omega}t)\cos(\bar{\omega}t)\right]^2 \\ =& \frac{1+a^2}{4} + \frac{1-a^2}{4}\cos(2\tilde{\omega}t) - \frac{1-a^2}{4}\cos(2\bar{\omega}t) \\ & - \frac{(1+a)^2}{8}\cos[2(\bar{\omega}-\tilde{\omega})t] - \frac{(1-a)^2}{8}\cos[2(\bar{\omega}+\tilde{\omega})t] \end{aligned} \tag{D.3}$$

with coefficients

$$\begin{aligned} \mathcal{O}(1): &\quad \frac{1+a^2}{4} \\ \mathcal{O}(\cos[2\tilde{\omega}t]): &\quad \frac{1-a^2}{4} \\ \mathcal{O}(\cos[2(\bar{\omega}+\tilde{\omega})t]): &\quad -\frac{(1-a)^2}{8} \\ \mathcal{O}(\cos[2\bar{\omega}t]): &\quad -\frac{1-a^2}{4} \\ \mathcal{O}(\cos[2(\bar{\omega}-\tilde{\omega})t]): &\quad -\frac{(1+a)^2}{8} \\ \mathcal{O}(\cos[2(\bar{\omega}-n\tilde{\omega})t]): &\quad 0 \quad \text{for} \quad |n| \geq 2\,. \end{aligned} \tag{D.4}$$

Ansatz for $h(t)$ for $a \geq 0$

$f(t)$ can be written as a product of the envelope $g(t)$ and a fast oscillating function $h(t)$. For $a > 0$, I make the ansatz

$$h(t) = \sum_{n=0}^{\infty} c_n \sin^2[(\bar{\omega} - n\tilde{\omega})t] \quad \text{with} \quad \sum_{n=0}^{\infty} c_n = 1 \tag{D.5}$$

and rewrite $g(t)$ and $h(t)$ in the same way as $f(t)$. $g(t)$ reads

$$g(t) = \frac{1+a^2}{2} + \frac{1-a^2}{2}\cos[2\tilde{\omega}t]\,. \tag{D.6}$$

The ansatz of $h(t)$ can be expressed through

$$h(t) = \frac{1}{2} - \frac{1}{2}\sum_{n=0}^{\infty} c_n \cos[2(\bar{\omega} - n\tilde{\omega})t]\,. \tag{D.7}$$

Inserting this into the ansatz for $f(t)$ reads

$$\begin{aligned} f(t) =& g(t)h(t) \\ =& \frac{1+a^2}{4} + \frac{1-a^2}{4}\cos[2\tilde{\omega}t] - \frac{1+a^2}{4}c_0\cos[2\bar{\omega}t] \\ & - \frac{1-a^2}{8}c_0\cos[2(\bar{\omega}+\tilde{\omega})t] - \frac{1-a^2}{8}c_1\cos[2\bar{\omega}t] \\ & - \frac{1+a^2}{4}\sum_{n=1}^{\infty} c_n \cos[2(\bar{\omega} - n\tilde{\omega})t] \\ & - \frac{1-a^2}{8}\sum_{n=1}^{\infty}[c_{n-1} + c_{n+1}]\cos[2(\bar{\omega} - n\tilde{\omega})t]\,. \end{aligned} \tag{D.8}$$

Sorting this into corresponding Fourier components results in

$$\begin{aligned} \mathcal{O}(1): &\quad \frac{1+a^2}{4} \\ \mathcal{O}(\cos[2\tilde{\omega}t]): &\quad \frac{1-a^2}{4} \\ \mathcal{O}(\cos[2(\bar{\omega}+\tilde{\omega})t]): &\quad -\frac{1-a^2}{8}c_0 \\ \mathcal{O}(\cos[2\bar{\omega}t]): &\quad -\frac{1+a^2}{4}c_0 - \frac{1-a^2}{8}c_1 \\ \mathcal{O}(\cos[2(\bar{\omega} - n\tilde{\omega})t]): &\quad -\frac{1+a^2}{4}c_n - \frac{1-a^2}{8}[c_{n-1} + c_{n+1}] \quad \text{for} \quad n \geq 1\,. \end{aligned} \tag{D.9}$$

The latter can be compared to the expansion coefficients from equation D.4. Terms of $\mathcal{O}(1)$ and $\mathcal{O}(\cos[2\tilde{\omega}t])$ already coincide. The $\mathcal{O}(\cos[2(\bar{\omega}+\tilde{\omega})t])$ terms lead to

$$c_0 = \frac{1-a}{1+a} \in [0,1] \quad \text{for} \quad 0 \leq a \leq 1\,. \tag{D.10}$$

Using this, the $\mathcal{O}(\cos[2\bar{\omega}t])$ terms lead to

$$c_1 = \frac{4a}{(1+a)^2} = 1 - c_0^2 \,. \tag{D.11}$$

Comparing the $\mathcal{O}(\cos[2(\bar{\omega}-\tilde{\omega})t])$ terms

$$-\frac{(1+a)^2}{8} \stackrel{!}{=} -\frac{1+a^2}{4}c_1 - \frac{1-a^2}{8}[c_0 + c_2] \,, \tag{D.12}$$

results in

$$c_2 = -\frac{4a(1-a)}{(1+a)^3} = -c_0 c_1 \,. \tag{D.13}$$

Comparing the $\mathcal{O}(\cos[2(\bar{\omega}-2\tilde{\omega})t])$ terms

$$0 \stackrel{!}{=} -\frac{1+a^2}{4}c_2 - \frac{1-a^2}{8}[c_1 - c_3] \tag{D.14}$$

results in

$$c_3 = \frac{(1-a)^2}{(1+a)^2}c_1 = c_0^2 c_1 \tag{D.15}$$

Therefore, I make the ansatz

$$c_n = -c_0 c_{n-1} = (-c_0)^{n-1} c_1 = (-c_0)^{n-1}(1-c_0^2) \quad \text{for} \quad n \geq 2 \,, \tag{D.16}$$

which I proof with mathematical induction.

I already showed that the latter is fulfilled for $c_2 = -c_0 c_1$ and $c_3 = c_0^2 c_1$. I assume that the ansatz is correct up to c_n and proof its correctness for $n+1$ using the $\mathcal{O}(\cos[2(\bar{\omega}-n\tilde{\omega})t])$ terms

$$0 \stackrel{!}{=} -\frac{1+a^2}{4}c_n - \frac{1-a^2}{8}[c_{n-1} - c_{n+1}] \,. \tag{D.17}$$

This results in

$$c_{n+1} = -\frac{1-a}{1+a}c_n = -c_0 c_n \tag{D.18}$$

and finishes the proof.

Moreover, the condition $\sum_{n=0}^{\infty} c_n = 1$ is automatically fulfilled: In the case of no coupling ($a = 1$), $c_1 = 1$ and the other coefficients vanish, which results in $f(t) = \sin^2[(\bar{\omega}-\tilde{\omega})t]$ with $\hbar(\bar{\omega}-\tilde{\omega}) = \bar{E} - E_0 + \Delta E = E_{\mathrm{AD}^*} - E_0$. As expected, this is an oscillation of the donor density with the donor excitation frequency, which independent of the acceptor. In the case of resonant coupling ($a = 0$), $c_0 = 1$ and the other coefficients vanish. This again results in the expected result $f(t) = \cos^2(\tilde{\omega}t)\sin^2(\bar{\omega}t)$ with $\hbar\tilde{\omega} = |V|$ of the donor-part from equation (5.16). In general, for $0 < a < 1$ and therefore $0 < c_0 < 1$

$$\begin{aligned}\sum_{n=0}^{\infty} c_n &= c_0 + c_1 + \sum_{n=2}^{\infty} c_1(-c_0)^{n-1} = c_0 + c_1\left[1 + \sum_{n=1}^{\infty}(-c_0)^n\right] \\ &= c_0 + \underbrace{(1-c_0^2)}_{=c_1}\underbrace{\sum_{n=0}^{\infty}(-c_0)^n}_{=\frac{1}{1+c_0}} = c_0 + (1-c_0) = 1 \,.\end{aligned} \tag{D.19}$$

Ansatz for $h(t)$ for $a \leq 0$

For $-1 \leq a \leq 0$, the proof is essentially the same if one uses the ansatz

$$h(t) = \sum_{n=0}^{\infty} c_n \sin^2[(\bar{\omega} + n\tilde{\omega})t] \quad \text{with} \quad \sum_{n=0}^{\infty} c_n = 1\,. \tag{D.20}$$

The only difference to the $a > 0$ case is the sign in front of $n\tilde{\omega}$ inside the sine. During the evaluation, this interchanges the roles of the $\mathcal{O}(\cos[2(\bar{\omega}-\tilde{\omega})t])$ and $\mathcal{O}(\cos[2(\bar{\omega}+\tilde{\omega})t])$ terms and leads to $c_0 = \frac{1+a}{1-a} \in [0,1]$.

Proof that $h(t)$ has amplitude one

At this stage, I showed that the time dependence of the squared density oscillation of the donor can be decomposed into a product of an envelope $g(t)$ and a fast oscillating function $h(t)$. In general, a function that is the sum of different Fourier components with different frequencies shows a kind of beat signal. In order to show that $g(t)$ is really the envelope of $f(t)$, one has to proof that $h(t)$ shows no beat but can be expressed through an oscillating function with a time-dependent frequency but a constant amplitude of one. For the limiting cases $a = 1$ and $a = 0$, this is easily fulfilled. In view of the representation of $h(t)$ in equation (D.7), it is sufficient to show (here again for $a > 0$) that

$$\begin{aligned} &\sum_{n=0}^{\infty} c_n \cos[2(\bar{\omega} - n\tilde{\omega})t] \stackrel{!}{=} \cos[\omega(t)t] \\ \Leftrightarrow &\frac{1}{2}\sum_{n=0}^{\infty} c_n e^{2i(\bar{\omega}-n\tilde{\omega})t} + c.c. \stackrel{!}{=} \frac{1}{2} e^{i\omega(t)t} + c.c. \end{aligned} \tag{D.21}$$

with $\omega(t) \in \mathbb{R}$ and $c.c.$ being the complex conjugate of the expression in front of it. This is fulfilled if

$$e^{i\omega(t)t} = \sum_{n=0}^{\infty} c_n e^{2i(\bar{\omega}-n\tilde{\omega})t}\,. \tag{D.22}$$

Application of the complex logarithm to this expression for $t > 0$ results in

$$\omega(t) = \frac{-i}{t} \ln\left[\sum_{n=0}^{\infty} c_n e^{2i(\bar{\omega}-n\tilde{\omega})t}\right] = \frac{-i}{t} \ln\left[e^{2i\bar{\omega}t} \sum_{n=0}^{\infty} c_n e^{-2ni\tilde{\omega}t}\right] \tag{D.23}$$

$$= 2\bar{\omega} - \underbrace{\frac{i}{t} \ln\left[\underbrace{\sum_{n=0}^{\infty} c_n e^{-2ni\tilde{\omega}t}}_{z}\right]}_{\nu}\,. \tag{D.24}$$

z and ν are complex numbers. ν can be expressed through the complex logarithm by $\nu = \ln(|z|) + i\arg(z)$ where $\arg(z)$ is the argument of z. $\omega(t)$ is real exactly if ν is

purely imaginary, which is true for $|z| = 1 \Leftrightarrow |z|^2 = zz^* = 1$. The latter reads

$$zz^* \overset{c_n \in \mathbb{R}}{=} \sum_{n=0}^{\infty}\sum_{n'=0}^{\infty} c_n c_{n'} e^{-2(n-n')i\tilde{\omega}t} = \underbrace{\sum_{n=0}^{\infty} c_n^2}_{\text{(I)}} + \underbrace{\sum_{n=0}^{\infty}\sum_{n'=n+1}^{\infty} c_n c_{n'} \overbrace{\left[e^{-2(n-n')i\tilde{\omega}t} + c.c.\right]}^{=2\cos[2(n-n')\tilde{\omega}t]}}_{\text{(II)}} . \tag{D.25}$$

The $n = n'$-part (I) reads (note again that $a > 0$ here)

$$\sum_{n=0}^{\infty} c_n^2 = c_0^2 + \underbrace{(1-c_0^2)^2}_{=c_1^2} \underbrace{\sum_{n=0}^{\infty} c_0^{2n}}_{\overset{c_0 \in [0,1[}{=} \frac{1}{1-c_0^2}} = c_0^2 + (1-c_0^2) = 1 . \tag{D.26}$$

The $n \neq n'$-part (II) can be sorted by its Fourier components

$$\begin{aligned}
2\sum_{n=0}^{\infty}\sum_{n'=n+1}^{\infty} c_n c_{n'} \cos[2(n-n')\tilde{\omega}t] &\overset{k=n'-n}{=} 2\sum_{n=0}^{\infty}\sum_{k=1}^{\infty} c_n c_{n+k} \cos[2k\tilde{\omega}t] \\
&= 2\sum_{k=1}^{\infty} \cos[2k\tilde{\omega}t] \left[\underbrace{c_0 c_k}_{=(-1)^{k-1}c_0^k c_1} + \sum_{n=1}^{\infty} c_n \underbrace{c_{n+k}}_{=(-c_0)^k c_n} \right] \\
&= 2\sum_{k=1}^{\infty} (-1)^k \cos[2k\tilde{\omega}t] \underbrace{\left[-c_0^k c_1 + c_0^k \underbrace{\sum_{n=1}^{\infty} c_n^2}_{\overset{\text{(I)}}{=} 1-c_0^2 = c_1} \right]}_{=0} \\
&= 0 .
\end{aligned} \tag{D.27}$$

Hence, $|z|^2 = \text{(I)} + \text{(II)} = 1$. This works analogously for $a < 0$. Thus, I proved that $h(t)$ is a fast oscillating function with a time-dependent, real-valued frequency $\omega(t)$ but an amplitude of one. $g(t)$ is therefore the envelope of $f(t)$ and shows the time-dependence of the donor energy from the donor-acceptor model presented in section 5.4.1 for a resonant and non-resonant coupling.

E. Numerical details and supporting information

In the following, I outline the fundamental numerical parameters for the calculations presented in the main text and show supporting material. I always checked the reliability of my calculations by compating to calculations with different numerical parameters.

E.1. Pseudo potentials

The norm-conserving Troullier-Martins pseudo potentials [TM91] that I used for carbon (C), nitrogen (N), oxygen (O), hydrogen (H), and sodium (Na) are well tested. Yet, for magnesium (Mg) a new one had to be generated[62] and tested. The tests are outlined in the following.

To test the Mg pseudo potential, I did ground state DFT calculations for different molecules using the real-space pseudo-potential code PARSEC [Kro+06] and compared the eigenvalues with all-electron basis-set calculations[63] using Q-Chem [Sha+15]. We did calculations for benzene and different variations of BChl*a*. Specifically, to sort out the influence of real-space grid and basis-set parameters, we considered a normal BChl*a*, a BChl*a* without the Mg atom, and the BChl*a* with the Mg atom replaced by a C atom. Each BChl*a* was once treated with the complete phytyl tail and with the phytyl tail replaced by H.

In Q-Chem we used the basis sets 6-31G, 6-31G(d,p), and 6-311G(d,p). In PARSEC we used different grid sizes, grid spacings ($0.3 - 0.4\,\mathrm{a_0}$ for BChl*a* and $0.125 - 0.3\,\mathrm{a_0}$ for benzene), convergence criteria, and pseudo potentials.

For all test sets, the eigenvalues around the HOMO differed by about $0.4 - 0.6\,\mathrm{eV}$ between PARSEC and Q-Chem with the 6-31G and 6-31G(d,p) basis sets. Since the deviation is significant, we tested the influence on the PARSEC parameters, especially the grid spacing. We found the results to be quite insensitive to these variations, especially for a grid spacing $\leq 0.3\,\mathrm{a_0}$. Finally, using a larger basis set 6-311G(d,p) changed the Q-Chem eigenvalues by up to $\approx 0.3\,\mathrm{eV}$, which closed most of the gap between Q-Chem and PARSEC results.

The differences between the Q-Chem and PARSEC eigenvalues of the original BChl*a* (i.e., with Mg pseudo potential) and the BChl*a* with Mg replaced by C (i.e., without Mg pseudo potential) were very close. Tests with different pseudo potentials for other atoms, specifically for C and H, had negligible influence.

I therefore conclude that the newly generated Mg pseudo potential with a cutoff radius of $2.56\,\mathrm{a_0}$ works well. This is also confirmed by the fact that changing the Q-Chem basis set had a much larger effect than using different pseudo potentials. The large effect of different Q-Chem basis sets is also exemplified by the B302 spectra in figure 5.3.

The cutoff radii of the pseudo potentials used for calculations throughout this thesis were: C ($1.09\,\mathrm{a_0}$), H ($1.39\,\mathrm{a_0}$), O ($1.10\,\mathrm{a_0}$), N ($0.99\,\mathrm{a_0}$), Mg ($2.56\,\mathrm{a_0}$), and Na ($3.09\,\mathrm{a_0}$). All pseudo potentials have been generated self-consistently with the LDA functional. As local component for the Kleinman-Bylander transformation [KB82; Kro+06], I

[62] Pseudo potentials were generated by Prof. Stephan Kümmel (University of Bayreuth, Germany).
[63] Q-Chem calculations were performed by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil).

chose the s-component ($l = 0$) for all atom types but for O, for which I chose the p-component ($l = 1$).

E.2. Presented calculations

E.2.1. Section 3

Polyacetylen The linear polyacetylen chain discussed in section 3.5 consists of 30 carbon atoms and 34 hydrogen atoms, which results in 77 occupied orbitals for one spin channel in a spin-unpolarized calculation. I did the calculation with 80 orbitals in total, i.e., three additional, virtual orbitals. In BTDFT the chain is aligned in z-direction, in TD-PARSEC it is aligned in x-direction such that the grid parallelization can be utilized in an optimal way in both programs. In BTDFT the half-axes of the grid were $(30, 30, 45)\,\mathrm{a_0}$. In TD-PARSEC, the large half-axis is aligned towards the x-direction. The grid spacing was $0.32\,\mathrm{a_0}$, which results in about $5.2 \cdot 10^6$ grid points. The system was initially excited with a boost as explained in section 4 with an energy of 0.01 Ry in z-direction. In BTDFT the tolerances for solving the Hartree potential and the Crank-Nicolson equation were 10^{-7} with the backward error as introduced in section 3.3.5. In TD-PARSEC, the tolerance for solving the Hartree potential cannot be specified in the configuration file. The time-step sizes and the parallelization used are discussed in the main text in section 3.5.

Note that the boost as chosen above is a small perturbation of the system. If the system is subject to a stronger excitation, the solution of the Crank-Nicolson equation can be more effort such that the performance drops down. This influences specifically the comparison between the BTDFT calculations with the Taylor and the Crank-Nicolson propagators.

Two bacteriochlorophylls The bacteriochlorophyll system used for the second performance test is the B302-B303 system discussed in section 5.3.2. It consists of 35 carbon atoms, 36 hydrogen atoms, six oxygen atoms, four nitrogen atoms, and one magnesium atom, which results in 234 occupied orbitals for one spin channel in a spin-unpolarized calculation. For the performance test I used in total 240 orbitals, i.e., six additional, virtual orbitals. The half-axes of the grid were $(34, 34, 27)\,\mathrm{a_0}$. The grid spacing was $0.18\,\text{Å} \approx 0.34\,\mathrm{a_0}$, which results in about $3.3 \cdot 10^6$. The system was initially excited with a boost as explained in section 4 with an energy of 0.01 Ry in z-direction. The type of excitation has no effect on the relative performance data as presented in section 3.5.2. In BTDFT the tolerance for solving the Hartree potential was 10^{-7}, the one for solving the Crank-Nicolson equation was 10^{-6} with the backward error as introduced in section 3.3.5. The time-step size used was 0.01 fs for a total propagation time of 0.3 fs for the tests. The computation times for initialization and IO are excluded in the data shown in section 3.5.2, which were generated with the current version of BTDFT. Thus, the presented data represent the time required for the pure propagation and justifies the short total propagation time.

E.2.2. Section 4

Remarks about numerical parameters In section 4 I showed that the electronic dipole spectra after a boost excitation consist of sine cardinal shaped lines. However,

this is only true if the numerical parameters are chosen properly. The following can happen:

- If the initial state is no proper ground state of the system, the line shapes will deviate from sine cardinals and additional artifacts can appear.
- If the convergence criteria for solving the Crank-Nicolson equation or Poisson's equation for the Hartree potential are too weak, the amplitude of the time-dependent dipole moment is not stable, i.e., the system shows some kind of self-enhancement or damping. In this case, the line shapes deviate from sine cardinals, e.g., their side minima are much deeper. Yet, I found the excitation energies usually well described.
- Finally, if the boost is too strong, the spectrum shows additional artifacts. If the boost is too weak, numerical noise disturbs the resulting spectrum.

The convergence criteria used for the Crank-Nicolson propagator and the Hartree solver can be chosen automatically by BTDFT (if a negative value is given in the configuration file, see appendix A.3.4). So far, I found the resulting criteria to be sufficient to get reliable results such as those shown in figure 4.1. The valid regime of the boost strength is quite large. For sodium systems, I usually used a boost energy of $E^{(\mathrm{boost})} = 0.0001\,\mathrm{Ry}$. For the BChl*a* systems, I typically used $E^{(\mathrm{boost})} = 0.01\,\mathrm{Ry}$.

Spectrum and transition densities of the sodium-4 cluster The Na_4 structure is described in figure 4.1 (a1). I used a grid spacing of $0.9\,a_0$[64] as in [VÖC99]. The boundary ellipsoid had half-axes of 25 a0, the time-step size was 0.01 fs, the boost strength was $E_{\mathrm{boost}} = 0.0001\,\mathrm{Ry}$. Propagation times and boost directions are described in the text.

Excitations above 3 eV were still sensitive to the grid size up to half-axes of about $50\,a_0$[65]. This might be due to the exponential decay of the TDLDA potential, which raises the potential well at the position of the finite system. The latter leads to excited states that are too weakly bound[66] and have an extent that reaches towards the boundary of the grid.

Spectrum and transition densities of the bacteriochlorophyll The BChl*a* structure is displayed in figure 4.1 (b1) and taken from [ONS10]. The original structure was rotated around the z-axis by $+45°$ and subsequently around the x-axis by $-45°$. This moves the bacteriochlorin ring into the xy-plane.

For the spectrum and the transition densities I used a grid spacing of $0.34\,a_0$. For a single BChl*a* I used a grid radius of $22\,a_0$, which I found to be sufficient for the description of the Q-band excitations through various tests with different grid sizes. Furthermore, I used a boost strength of $E_{\mathrm{boost}} = 0.01\,\mathrm{Ry}$, a propagation time of 100 fs for the spectrum, and 50 fs for the transition densities. I further discuss the influence of different parameters by means of B302 in appendix E.3.1.

[64] For a better quality, a grid spacing of $0.7 - 0.8\,a_0$ should be used.
[65] Confirmed by Sebastian Hammon, group of Prof. Stephan Kümmel (University of Bayreuth).
[66] This effect also leads to the difficulty in computing electron affinities as noted by [Cap06, Footnote 15].

E.2.3. Section 5

Tuning the range separation parameter The tuning procedure was done by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil) along with all Q-Chem calculations. I extracted the following details from a compilation of his results.

The range separation parameter ω was tuned such that the ionization potentials $IP(i)$ of the neutral ($i = N$) and anionic ($i = N + 1$) systems as calculated from DFT total energy differences agree as close as possible with the respective HOMO eigenvalues $\varepsilon_{\mathrm{HOMO}}(N)$ and $\varepsilon_{\mathrm{HOMO}}(N + 1)$ [SKB09b; SKB09a; Kar+11] (compare to equation (2.17)). The tuning was done only for B302 and the 6-31G(d,p) basis set for smaller atoms and EPC-LANL2DZ for magnesium. This resulted in $\omega = 0.171\,\mathrm{a_0^{-1}}$ for ωPBE. The corresponding values for ionization energies and HOMO eigenvalues are $IP(N) = 5.826\,\mathrm{eV}$, $\varepsilon_{\mathrm{HOMO}}(N) = -5.836\,\mathrm{eV}$, $IP(N + 1) = 1.601\,\mathrm{eV}$, and $\varepsilon_{\mathrm{HOMO}}(N + 1) = -1.575\,\mathrm{eV}$. The calculations on B303 and the combined B302-B303 system used the same ω.

Real-time spectra of B850 bacteriochlorophylls The presented spectra were obtained with a propagation time of typically 100 fs and 200 fs for the coupled B302-B303 system. Other numerical parameters for the different calculations are discussed in the main text in section 5.3 and in appendix E.3.1.

General Coulomb coupling strengths To compute V^{Coul} between two transition densities ρ_1 and ρ_2, which are both given on a real-space grid, I compute the energy of one transition density in the Hartree potential that is generated by the second transition density

$$V^{\mathrm{Coul}} = \int \rho_1(\boldsymbol{r}) \underbrace{\left[\int \frac{e^2}{4\pi\epsilon_0} \frac{\rho_2(\boldsymbol{r'})}{|r - r'|} d^3r' \right]}_{=v_H[\rho_2](\boldsymbol{r})} d^3r\,. \tag{E.1}$$

To this end, I extend the grid of ρ_2 to enclose the grid of ρ_1 and calculate the Hartree potential $v_{\mathrm{H}}[\rho_2]$ by solving Poisson's equation on the extended grid. Since the grid points of the real-space grids of both transition densities do in general not match, I use linearly interpolated values of $v_{\mathrm{H}}[\rho_2]$ to evaluate the r-integral on the real-space grid of ρ_1.

Coupling strengths between sodium dimers The single Na_2 calculations were done on a grid with a spacing of $0.8\,\mathrm{a_0}$ and half-axes of $25\,\mathrm{a_0}$ perpendicular and $35\,\mathrm{a_0}$ parallel to the molecular axis. For the Na_2-Na_2 calculations I used the same grid but increased the radius along the inter-dimer axis to $45\,\mathrm{a_0}$ for inter-dimer distances above $20\,\mathrm{a_0}$. At the distance of $20\,\mathrm{a_0}$ between the dimers I checked the consistency of the results from calculations with both grid sizes. For all calculations, I used a time-step size of 0.01 fs, a propagation time of 50 fs for spectrum and transition densities, and a boost strengths of $E_{\mathrm{boost}} = 0.00005\,\mathrm{Ry}$ in one half-space.

Coupling strengths between bacteriochlorophylls I used the $\mathrm{Q_y}$ transition densities and excitation energies from the single BChl*a* calculations as presented in section 4.

The structure of the original BChl$a$ was rotated by 45° around the z-axis and subsequently by −45° around the x-axis. After this transformation, the bacteriochlorin ring of the BChl$a$ lies approximately in the xy-plane with the Q_y transition dipole along $(0.4497, -0.8887, -0.0897)$ as predicted from real-time TDLDA. For the Davydov splitting calculation with an inter-BChl$a$ distance of 8 Å (along the z-axis), I primarily boosted the Q_y transition of the donor ($\propto (1, -1, 0)$), with 0.005 Ry in the donor half-space.

The donor and acceptor dipole moments were analyzed separately and lead to equal results for the splitting of the Q_y lines. The fit of two lines to the data was perfectly possible in this case, even if BChl$a$ is no real two-level system (but treated as one). However, the heights of the lines resulting from symmetric and anti-symmetric coupling differed in contradiction to the expectation from section 5.4.3. For these calculations, I used the same grid spacing as for a single BChl$a$ and an ellipsoidal grid with half-axes of 28 a_0 in x- and y- directions and 35 a_0 along z. The propagation time was 100 fs.

E.3. Additional calculations

E.3.1. Additional spectra of B301, B302, and B303

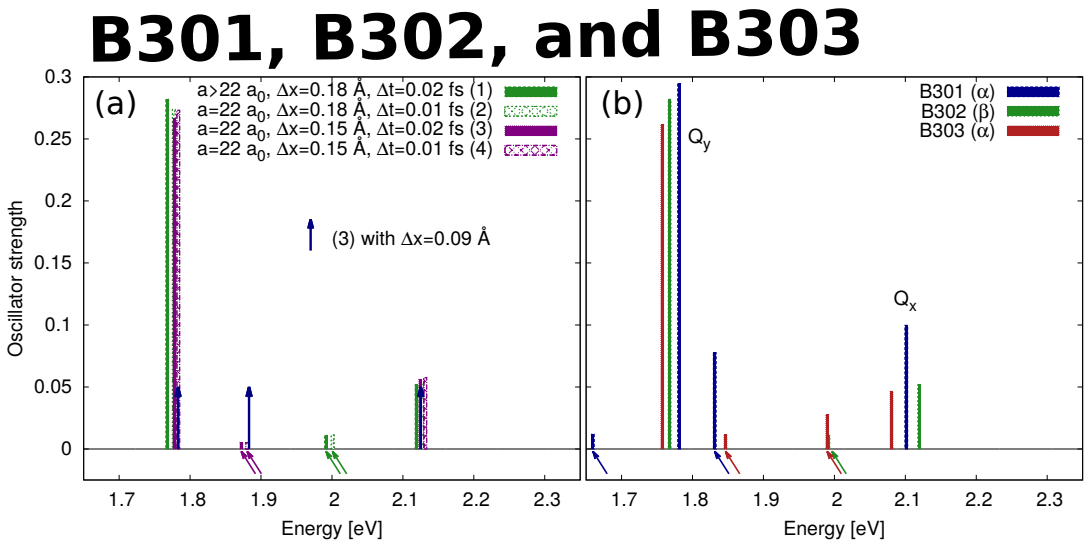
I checked the reliability of the B302 spectra that I presented in section 5.3 by varying relevant numerical parameters: The grid spacing ($\Delta x = 0.15$ Å, $\Delta x = 0.18$ Å, $\Delta x = 0.09$ Å), the grid size with half-axes $\boldsymbol{a}$ (typically $\boldsymbol{a} = (22, 22, 22)\,a_0$ or $\boldsymbol{a} = (25, 25, 27)\,a_0$), and the time-step size ($\Delta t = 0.01$ fs or $\Delta t = 0.02$ fs). The results for B302 are shown in figure 5.4 (a).

Data set (1) and (3) are the ones shown in the main part of the thesis. Data set (3) was calculated with half-axes of $\boldsymbol{a} = (22, 22, 22)\,a_0$ while (1) had a larger grid with $\boldsymbol{a} = (25, 25, 27)\,a_0$. The Q_y and Q_x transitions are stable under variation of all numerical parameters.

Decreasing the time-step size from 0.02 fs to 0.01 fs causes a blue-shift of the spectral lines by ≈ 0.01 eV, which can be attributed to the time-discretization error. The different grid sizes of (1) and (2) do not change this such that the grid size of $\boldsymbol{a} = (22, 22, 22)\,a_0$ is sufficient.

As already mentioned in section 5.3, the spurious charge-transfer states (CT1 and CT2) react sensitive on the grid spacing (see (1) vs. (3) and (2) vs. (4)). Decreasing the grid spacing from 0.18 Å to 0.15 Å shifts the charge-transfer states by ≈ 0.12 eV down in energy. The effect on the Q_y and Q_x transitions is again ≈ 0.01 eV. Decreasing the grid spacing further to 0.09 Å but keeping the other parameters from data set (3) (indicated by the arrows, oscillator strengths are not available) gives no further improvement. A grid spacing of $\Delta x = 0.15$ Å is therefore sufficient to describe all of the observed states, a grid spacing of $\Delta x = 0.18$ Å is sufficient for the Q-band excitations.

The geometry of different BChls in the crystal structure is slightly different, even for two α-BChls such as B301 and B303. Figure E.1 (b) shows the real-time spectra of B301, B302, and B303 as calculated with $\Delta x = 0.18$ Å and $\Delta t = 0.02$ fs. While the Q-band excitations are quite close to each other, there are major differences in the spurious states that depend on these structural differences.



Data (a)

	RT-TDLDA				
Excit.	$\Delta x = 0.18$ Å		$\Delta x = 0.15$ Å		$\Delta x = 0.09$ Å
state	$\Delta t = 0.02$ fs	$\Delta t = 0.01$ fs	$\Delta t = 0.02$ fs	$\Delta t = 0.01$ fs	$\Delta t = 0.02$ fs
S_1	1.768 (0.282)	1.777 (0.274)	1.778 (0.266)	1.784 (0.273)	1.7831 (–)
S_2	1.992 (0.011)	2.001 (0.011)	1.872 (0.005)	1.881 (0.005)	1.8831 (–)
S_3	2.119 (0.052)	2.128 (0.052)	2.125 (0.056)	2.132 (0.058)	2.1253 (–)

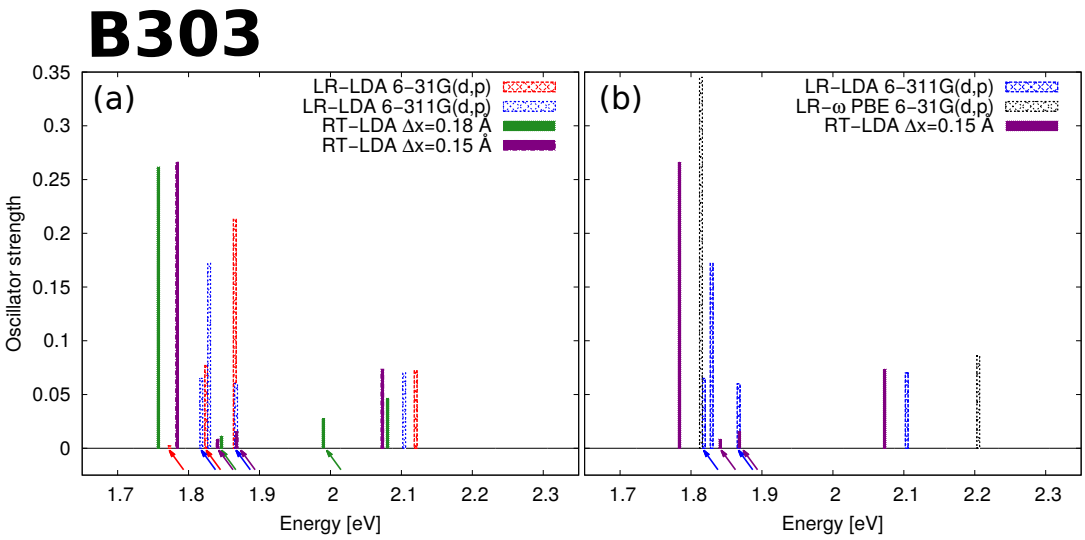
Data (b)

	RT-TDLDA, $\Delta x = 0.18$Å		
Excit.	**B301**	**B302**	**B303**
state			
S_1	1.660 (0.012)	1.768 (0.282)	1.757 (0.262)
S_2	1.781 (0.294)	1.992 (0.011)	1.847 (0.011)
S_3	1.831 (0.078)	2.119 (0.052)	1.990 (0.028)
S_4	2.101 (0.100)	–	2.081 (0.046)

Figure E.1: Real-time TDLDA singlet spectra of (a) B302 with different grid spacings and time-step sizes and of (b) B301, B302, and B303. Excitation energies are in eV, oscillator strengths in parenthesis.

B302 shows only one state between Q_y and Q_x, which can be identified with the CT2 state with a high transition density on the pyrrol rings (III) and (V). B303 shows the CT1 and CT2 states between Q_y and Q_x. B301 shows one charge-transfer state below Q_y (CT2) and one between Q_y and Q_x (CT1) for these numerical parameters. Their BTDFT transition densities are shown in figures E.3, E.10, and E.11 below.

To complete the study of the B302-B303 system, the real-time and Q-Chem spectra of B303 are displayed in figure E.2 in the way of the B302 spectra in figure 5.3. The main results are the same: The Q-band excitations of B303 from real-time TDLDA are quite stable with respect to variations of the grid spacing while two spurious states are seen at different energies around the Q_y excitation. The Q-Chem TDLDA result is again highly sensitive to basis-set variations, which leads to several states with non-vanishing oscillator strength in the Q_y region. The ωPBE calculation with the 6-31G(d,p) basis set shows exactly one Q_y and one Q_x transition whereas the latter is again shifted towards higher energies with respect to the TDLDA results. Real-time TDLDA and Q-Chem ωPBE transition densities are shown in figures E.11 and E.12.



Data

	RT-TDLDA		LR-TDLDA		LR-ωPBE
Excit. state	$\Delta x = 0.18$ Å	$\Delta x = 0.15$ Å	6-31G(d,p)	6-311G(d,p)	6-31G(d,p)
S_1	1.757 (0.262)	1.784 (0.266)	1.773 (0.002)	1.818 (0.065)	1.814 (0.345)
S_2	1.847 (0.011)	1.841 (0.008)	1.825 (0.077)	1.829 (0.172)	2.205 (0.086)
S_3	1.990 (0.028)	1.868 (0.016)	1.865 (0.213)	1.867 (0.060)	–
S_4	2.081 (0.046)	2.073 (0.074)	2.120 (0.072)	2.104 (0.070)	–

Figure E.2: Real-time (RT, solid) TDLDA and Q-Chem linear-response (LR, dashed) TDLDA and ωPBE singlet spectra for B303 with different grid spacings or basis sets. (a) TDLDA results with different parameters, (b) best TDLDA results compared to LR-ωPBE. Energies are in eV, strengths in parenthesis. Q-Chem calculations were performed by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil).

E.3.2. Transition densities and natural transition orbitals

The transition densities and NTOs of some excitations discussed in this thesis, specifically of B302, B303, and the combined B302-B303 system, are displayed in the figures below. Their contents are described in their captions and referred to in sections 5.3 and E.3.1. The states are labeled by their energetic order through S_n and additionally by their character, e.g., Q_y or CT1 in parenthesis. The Q-Chem calculations and NTO figures were generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). The latter are printed with his permission. All transition densities are taken at the same iso-surface of $\pm 0.0002\,a_0^{-3}$.

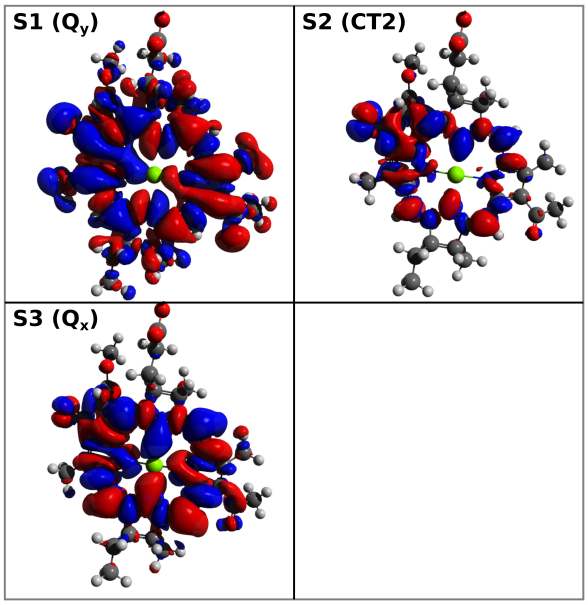


Figure E.3: Transition densities of the states S_1-S_3 of B302 from real-time TDLDA with $\Delta x = 0.18\,\text{Å}$.

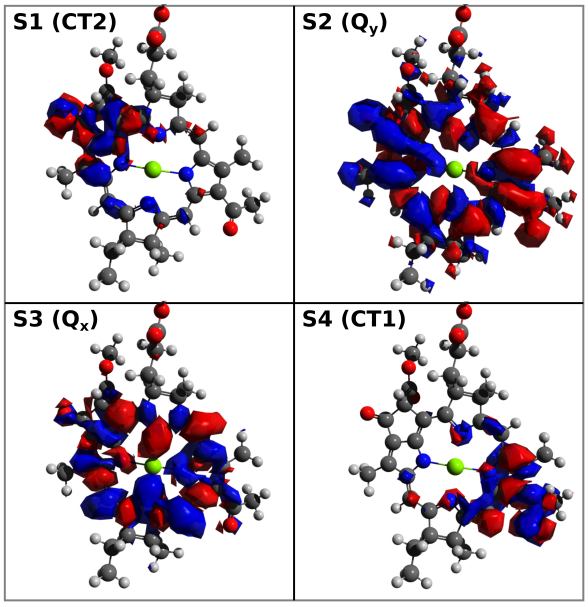


Figure E.4: Transition densities of the states S_1-S_4 of B302 from Q-Chem TDLDA with basis set 6-31G(d,p). Calculation done by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil).

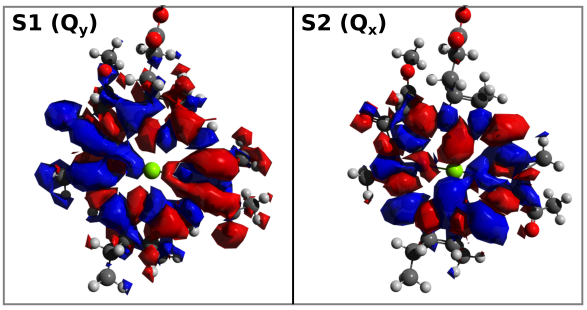


Figure E.5: Transition densities of the states S_1-S_2 of B302 from Q-Chem ωPBE with basis set 6-31G(d,p). Calculation done by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil).

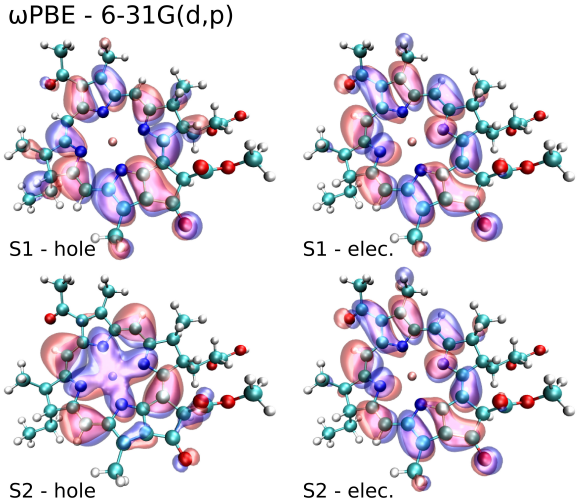


Figure E.6: NTOs of the states S_1-S_2 of B302 from Q-Chem ωPBE with basis set 6-31G(d,p). Graphic generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). Printed with permission.

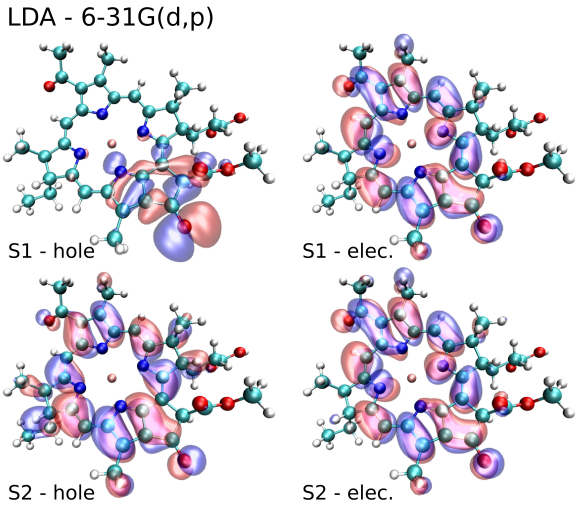


Figure E.7: NTOs of the states S_1-S_2 of B302 from Q-Chem TDLDA with basis set 6-31G(d,p). Graphic generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). Printed with permission.

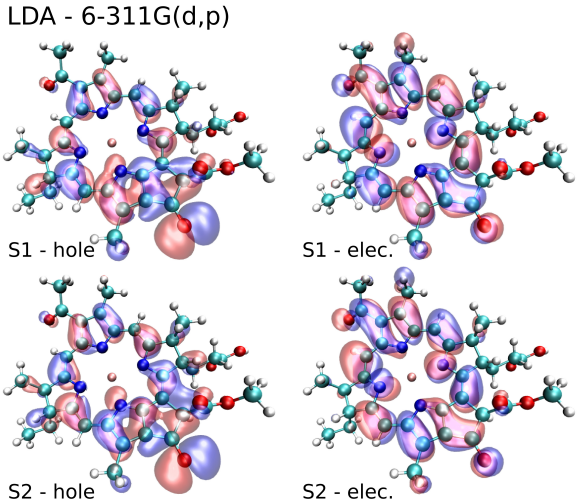


Figure E.8: NTOs of the states S_1-S_2 of B302 from Q-Chem TDLDA with basis set 6-311G(d,p). Graphic generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). Printed with permission.

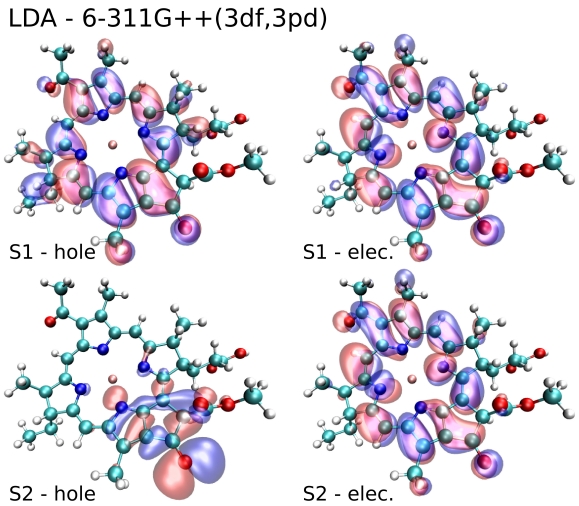


Figure E.9: NTOs of the states S_1-S_2 of B302 from Q-Chem TDLDA with basis set 6-311++G(3df,3pd). Graphic generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). Printed with permission.

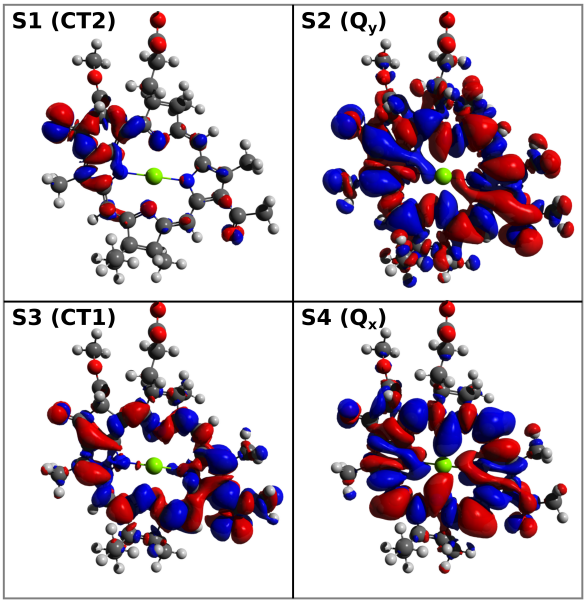


Figure E.10: Transition densities of the states S_1-S_4 of B301 from real-time TDLDA with $\Delta x = 0.18\,\text{Å}$.

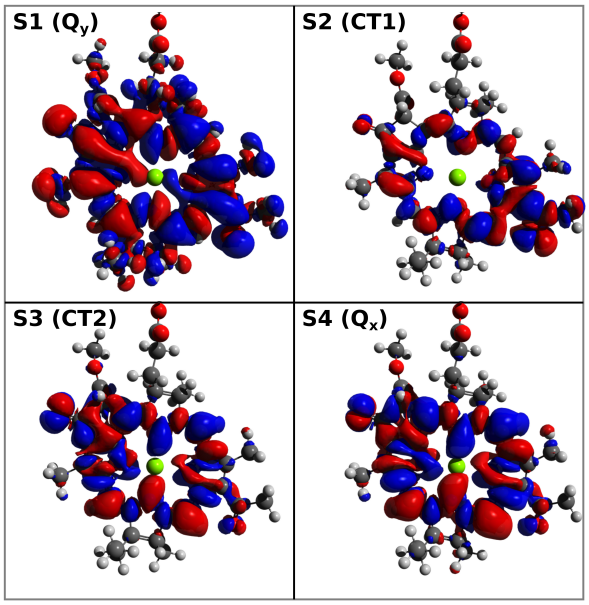


Figure E.11: Transition densities of the states S_1-S_4 of B303 from real-time TDLDA with $\Delta x = 0.18\,\text{Å}$.

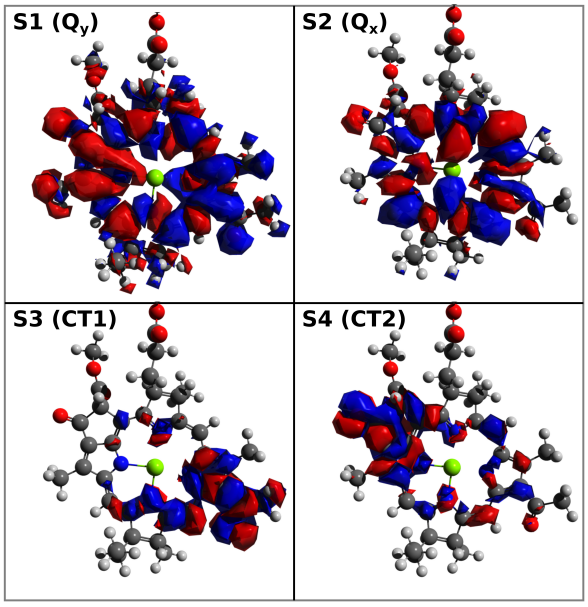


Figure E.12: Transition densities of the states S_1-S_2 of B303 from Q-Chem ωPBE with basis set 6-31G(d,p). Calculation done by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil).

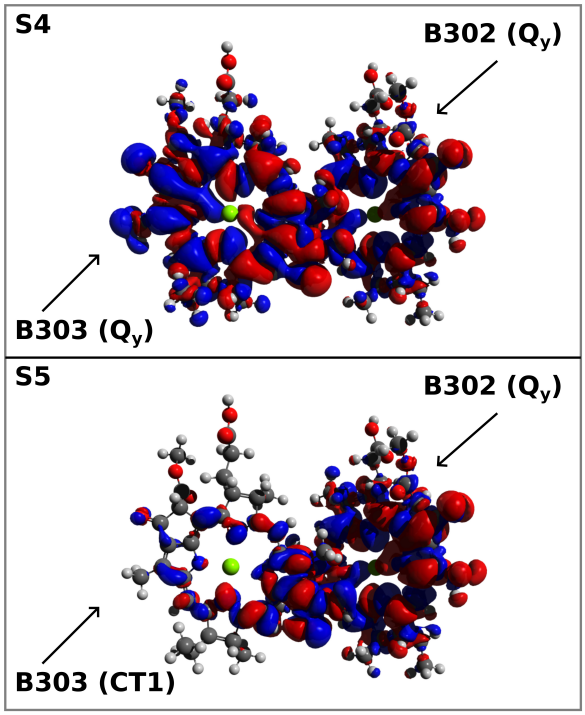


Figure E.13: Transition densities of the states S_4-S_5 of the B302-B303 system from real-time TDLDA with $\Delta x = 0.18\,\text{Å}$.

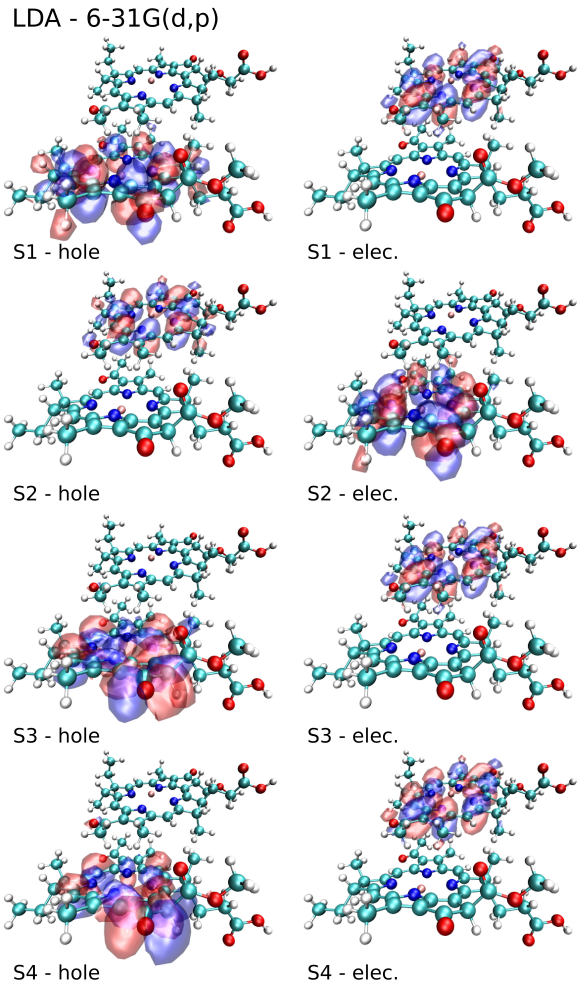


Figure E.14: NTOs of the states S_1-S_4 of the B302-B303 system from Q-Chem TDLDA with basis set 6-31G(d,p). Graphic generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). Printed with permission.

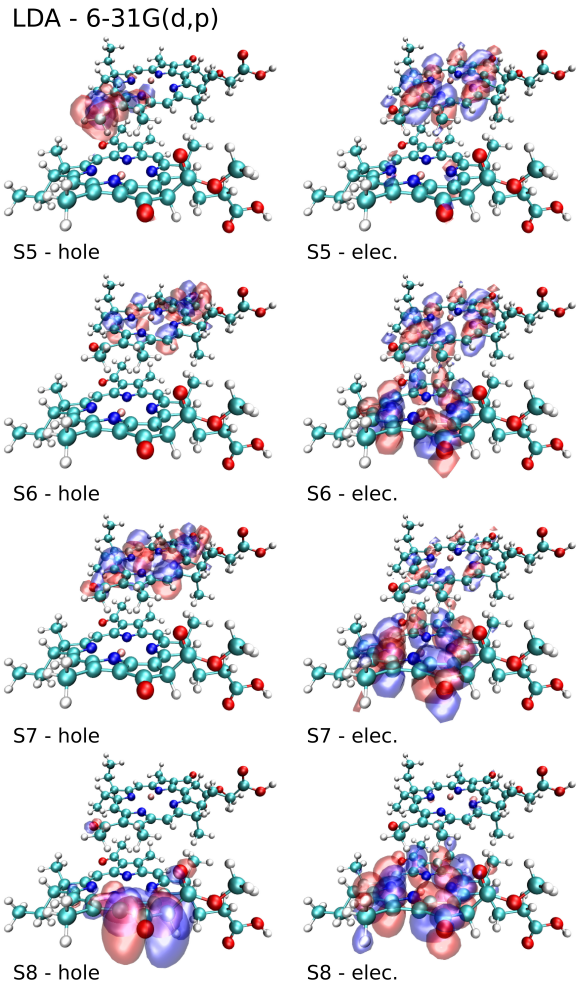


Figure E.15: NTOs of the states S_5-S_8 of the B302-B303 system from Q-Chem TDLDA with basis set 6-31G(d,p). Graphic generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). Printed with permission.

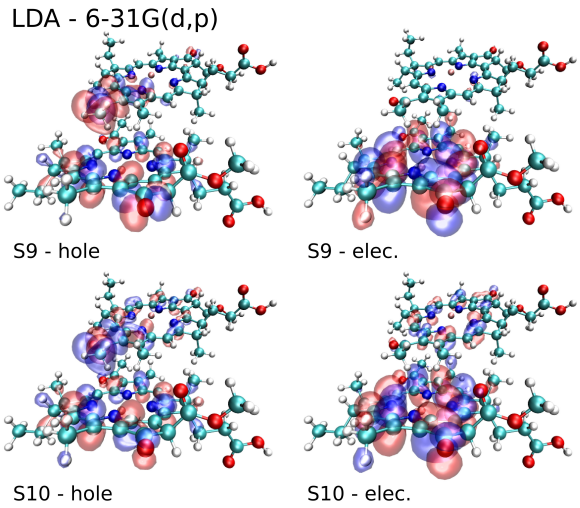


Figure E.16: NTOs of the states S_9-S_{10} of the B302-B303 system from Q-Chem TDLDA with basis set 6-31G(d,p). Graphic generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). Printed with permission.

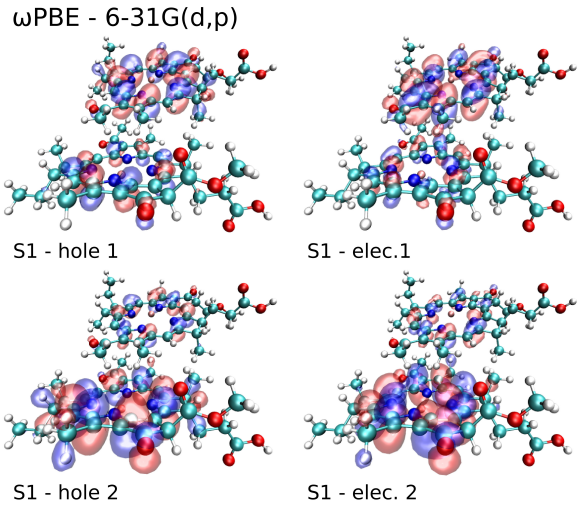


Figure E.17: NTOs of the states S_1-S_2 of the B302-B303 system from Q-Chem ωPBE with basis set 6-31G(d,p). Graphic generated by Prof. Thiago Branquinho de Queiroz (Federal university of ABC, Brasil). Printed with permission.

F. Preparation of the LH2 structure and the environment potential

I briefly outline the preparation of the LH2 structure as used in the context of the calculations done on the B850 ring during my work. The work in this section has been discussed with Prof. Matthias Ullmann (Computational biochemistry, University of Bayreuth) and Johannes Förster in his group and was mostly done by Johannes. I refer to him again in the sections below.

F.1. Preparation of the LH2 structure

The structure of the LH2 complex of *Rbl. acidophilus* (Strain 10050) was determined in the first place in 1995 by [McD+95] and again by [Pap+03] with a resolution of 2.0 Å and new structural features. It is explained in great detail in the literature [Fre+96; Pap+03; Hu+02; CGK06]. The structure that I used throughout shows the same structural features and was determined by Dr. Alastair Gardiner [GC] from the group of Prof. Richard Cogdell (Institute of Molecular Cell and Systems Biology, University of Glasgow) with a better resolution of 1.85 Å. The preparation of the LH2 structure for the TDDFT calculations was done by Johannes Förster and is outlined in the following.

The hydrogen atoms are usually not resolved by X-ray structure analysis and are therefore not included in the crystal structure data. They were added subsequently by minimizing the total energy with the CHARMM [Bro+83; Mac+98] force field while keeping the position of non-hydrogen atoms fixed. Since the protonation state of some residues depends on the pH-value, the temperature, and the respective environment, the overall protonation state [UB14; UU12] was checked for anomalies under standard (in-vivo) conditions.

F.2. Truncation of phytyl tails

The BChl*a* molecules in the LH2 complex have a phytyl tail, which is attached to the bacteriochlorin ring by an ester group (see figures 5.1 and 5.2). The phytyl tail plays an important, yet mostly structural role in that is anchors the BChl*a* in the protein scaffold [Fre+96]. Therefore, the phytyl tail is not relevant for the present electronic structure simulations and was truncated by replacing it with a hydrogen atom.

I added the hydrogen atom in the direction of the first carbon atom of the phytyl tail but with the respective $C - H$ bond length (0.975 Å) inside resulting carboxyl group. Replacing the phytyl tail by a methyl group instead of a hydrogen atom had negligible influence.

F.3. Preparation of histidine residues

The central magnesium atom (or Mg^{2+} ion) of a B850 BChl is coordinated by a histidine residue (see figures 5.1 (b) and 5.2 (b)), which distorts the bacteriochlorin ring. The histidine is therefore included directly in some of the TDDFT calculations. This can have an effect on the details of the intermolecular coupling and the energy transfer within the B850 ring. [CGK06, §3.2]

The respective histidine amino acids are integrated in the protein backbone of the respective α- of β-apoproteins inside the LH2 complex. In order to include them into the TDDFT calculation, I truncated the residues from their protein backbones. I did this between the C_α atom, which is part of the protein backbone, and the C_β, which is the first carbon atoms in the residue. The residue was saturated by a hydrogen atom in the same way as the phytyl tail, i.e., from the perspective of C_β in direction of the C_α with the bond length of a $C-H$ single bond (1.09 Å).

The protonation of the histidine residue (seen in figure 5.2 (b)) was chosen such that the nitrogen next to the BChl*a*'s central magnesium atoms was not protonated. The second nitrogen in the aromatic histidine ring was protonated. The distance between the central magnesium atom and the nearest nitrogen atom from the histidine residue is comparable to the distances between the magnesium atom and the nitrogen atoms within the bacteriochlorin ring.

F.4. The environment potential

The protein and solvent environment as well as the chromophores that were not simulated within TDDFT were modeled by an electrostatic environment potential by Johannes Förster. The environment was modeled as a dielectric medium with a dielectric constant of $\varepsilon_r = 80$ for the surrounding water (with a salt concentration of 0.15 mol/L), $\varepsilon_r = 4$ inside the protein complex, and $\varepsilon_r = 1$ at the position of the directly simulated molecules at a temperature of $T = 298\,\mathrm{K}$. The environment potential consist of two parts:

Protein potential The protein (and solvent) potential is generated by the partial charges that are assigned to the atoms in the surrounding environment and free charges in the surrounding solvent. The partial charges on the atoms of the amino acids in the protein are given by standard values [Bro+83; Mac+98]. The partial charges on the atoms of a B850 BChl were determined by CHELPG according to [BW90]. The charge density of the BChl*a* was determined by DFT calculations with the B3LYP functional [Bec93; Ste+94]. The BChl*a* was refined in two steps with the Def2-SV(P) and subsequently with the Def2-TZVP basis set using ORCA [Nee12]. The protein potential was then calculated by solving the Poisson-Boltzmann equation [Dol+07; Dol+04; Bak+01].

Reaction field The part of the LH2 complex that is directly simulated with TDDFT is not included in the protein potential. Still, it influences the environment. The reaction field displays the polarization in the environment that is generated by the simulated molecules. It is the difference between the protein potential as calculated above, once with and once without the directly simulated molecules. [Li+98; UB14]

The final environment potentials were given on an equidistant real-space grid with a grid spacing of 0.18 Å.

List of Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
ARPES	Angle-resolved photoemission spectroscopy
BChl	Bacteriochlorophyll
BiCG	Biconjugate gradient
BTDFT	The TDDFT program developed in this work
ccNUMA	Cache coherent non-uniform memory access
CG	Conjugate gradient
CG-SYM	Conjugate-gradient-like solver for complex-valued, symmetric matrices
CPU	Central processing unit
DC	Double complex
DFT	Density functional theory
DP	Double precision
DSF	Dipole strength function
EET	Excitation-energy transfer
EXX	Exact exchange
FFT	Fast Fourier transform
Flop	Floating point operation
FPU	Floating point unit
GGA	Generalized gradient approximation
GKS	Generalized Kohn-Sham
HK	Hohenberg-Kohn
HOMO	Highest occupied molecular orbital
IC	Interconnection (network)
IO	Input and output
KLI	Krieger, Li, and Iafrate
KS	Kohn-Sham
LDA	Local density approximation
LHF	Localized Hartree-Fock
Lup	Lattice site update
MPI	Message passing interface
NTO	Natural transition orbital
OEP	Optimized effective potential
OMB	Ohio state university (OSU) Micro Benchmarks
PBE	Perdew, Burke, and Ernzerhof
PSD	Power spectral density
RC	Reaction center
Rbl.	Rhodoblastus
RG	Runge-Gross
SCF	Self-consistent field
SIC	Self-interaction correction
SMT	Simultaneous multithreading
SRE	Self-consistency residual error

TDDFT	Time-dependent density functional theory
TDKS	Time-dependent Kohn-Sham
TDLDA	Time-dependent local density approximation
UBT	University of Bayreuth
UMA	Uniform memory access
xc	Exchange-correlation

Bibliography

[Ada+09] J. C. Adams, W. S. Brainerd, R. a. Hendrickson, R. E. Maine, J. T. Martin, and B. T. Smith. *The Fortran 2003 Handbook*. London: Springer London, 2009, page 713. ISBN: 978-1-84628-378-9 (cited on pages 13, 103).

[AB85] C.-O. Almbladh and U. von Barth. "Exact results for the charge and spin densities, exchange-correlation potentials, and density-functional eigenvalues". In: *Phys. Rev. B* 31.6 (1985), pages 3231–3244 (cited on page 6).

[AHD16] A. Anda, T. Hansen, and L. De Vico. "Multireference Excitation Energies for Bacteriochlorophylls A within Light Harvesting System 2". In: *J. Chem. Theory Comput.* 12.3 (2016), pages 1305–1313 (cited on page 55).

[And65] D. G. Anderson. "Iterative Procedures for Nonlinear Integral Equations". In: *J. ACM* 12.4 (1965), pages 547–560 (cited on page 25).

[And+15] X. Andrade, D. Strubbe, U. De Giovannini, A. H. Larsen, M. J. T. Oliveira, J. Alberdi-Rodriguez, A. Varas, I. Theophilou, N. Helbig, M. J. Verstraete, L. Stella, F. Nogueira, A. Aspuru-Guzik, A. Castro, M. A. L. Marques, and A. Rubio. "Real-space grids and the Octopus code as tools for the development of new simulation approaches for electronic systems". In: *Phys. Chem. Chem. Phys.* 17.47 (2015), pages 31371–31396 (cited on pages 1, 13, 25, 43).

[BLS10] R. Baer, E. Livshits, and U. Salzner. "Tuned Range-Separated Hybrids in Density Functional Theory". In: *Annu. Rev. Phys. Chem.* 61.1 (2010), pages 85–109 (cited on pages 9, 11, 60).

[BN05] R. Baer and D. Neuhauser. "Density Functional Theory with Correct Long-Range Asymptotic Behavior". In: *Phys. Rev. Lett.* 94.4 (2005), page 043002 (cited on pages 9, 11).

[Bak+01] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon. "Electrostatics of nanosystems: application to microtubules and the ribosome." In: *Proc. Natl. Acad. Sci. U. S. A.* 98.18 (2001), pages 10037–41 (cited on page 148).

[BH72] U. von Barth and L. Hedin. "A local exchange-correlation potential for the spin polarized case: I". In: *J. Phys. C Solid State Phys.* 5.13 (1972), pages 1629–1642 (cited on page 8).

[Bec93] A. D. Becke. "A new mixing of Hartree–Fock and local density-functional theories". In: *J. Chem. Phys.* 98.2 (1993), page 1372 (cited on pages 9, 11, 148).

[Ber16] F. Bertoldo. "Berechnung von Energietransfer in molekularen Systemen". Bachelor's thesis. University of Bayreuth, 2016 (cited on page 13).

[BW90] C. M. Breneman and K. B. Wiberg. "Determining atom-centered monopoles from molecular electrostatic potentials. The need for high sampling density in formamide conformational analysis". In: *J. Comput. Chem.* 11.3 (1990), pages 361–373 (cited on page 148).

[Bro+04] R. A. Broglia, G. Col'o, G. Onida, and E. Roman. *Solid State Physics of Finite Systems*. Edited by R. A. Broglia. Springer-Verlag Berlin Heidelberg, 2004 (cited on pages 43, 48).

[Bro+83] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. "CHARMM: A program for macromolecular energy, minimization, and dynamics calculations". In: *J. Comput. Chem.* 4.2 (1983), pages 187–217 (cited on pages 147, 148).

[Bur+03] W. Burdick, Y. Saad, L. Kronik, I. Vasiliev, M. Jain, and J. R. Chelikowsky. "Parallel implementation of time-dependent density functional theory". In: *Comput. Phys. Commun.* 156.1 (2003), pages 22–42 (cited on page 27).

[Bur07] K. Burke. "The abc of dft". In: *Dep. Chem. Univ. Calif.* (2007) (cited on pages 3, 4).

[Cai+06] Z.-L. Cai, M. J. Crossley, J. R. Reimers, R. Kobayashi, and R. D. Amos. "Density Functional Theory for Charge Transfer: The Nature of the N-Bands of Porphyrins and Chlorophylls Revealed through CAM-B3LYP, CASPT2, and SAC-CI Calculations". In: *J. Phys. Chem. B* 110.31 (2006), pages 15624–15632 (cited on pages 50, 60).

[CSR02] Z.-L. Cai, K. Sendt, and J. R. Reimers. "Failure of density-functional theory and time-dependent density-functional theory for large extended π systems". In: *J. Chem. Phys.* 117.12 (2002), page 5543 (cited on page 60).

[CRS97] F. Calvayrac, P. Reinhard, and E. Suraud. "Spectral Signals from Electronic Dynamics in Sodium Clusters". In: *Ann. Phys. (N. Y).* 255.1 (1997), pages 125–162 (cited on pages 1, 25, 43–45).

[Cal+00] F. Calvayrac, P.-G. Reinhard, E. Suraud, and C. Ullrich. "Nonlinear electron dynamics in metal clusters". In: *Phys. Rep.* 337.6 (2000), pages 493–578 (cited on pages 9, 25, 26, 43–45).

[Cap06] K. Capelle. "A bird's-eye view of density-functional theory". In: *Brazilian J. Phys.* 36.4a (2006), pages 1318–1343 (cited on pages 3–6, 9, 10, 67, 133).

[CAS95] M. E. CASIDA. "Time-Dependent Density Functional Response Theory for Molecules". In: *Recent Adv. Density Funct. Methods, Vol. 1*. Volume 9. 1995, pages 155–192 (cited on pages 6, 43, 50).

[Cas+98] M. E. Casida, C. Jamorski, K. C. Casida, and D. R. Salahub. "Molecular excitation energies to high-lying bound states from time-dependent density-functional response theory: Characterization and correction of the time-dependent local density approximation ionization threshold". In: *J. Chem. Phys.* 108.11 (1998) (cited on pages 43, 50).

[Cas+06] A. Castro, H. Appel, M. Oliveira, C. A. Rozzi, X. Andrade, F. Lorenzen, M. A. L. Marques, E. K. U. Gross, and A. Rubio. "octopus: a tool for the application of time-dependent density functional theory". In: *Phys. status solidi* 243.11 (2006), pages 2465–2488 (cited on pages 13, 25, 43).

[CMR04] A. Castro, M. A. L. Marques, and A. Rubio. "Propagators for the time-dependent Kohn–Sham equations". In: *J. Chem. Phys.* 121.8 (2004), pages 3425 (cited on pages 1, 25, 26, 43).

[CA80] D. M. Ceperley and B. J. Alder. "Ground State of the Electron Gas by a Stochastic Method". In: *Phys. Rev. Lett.* 45.7 (1980), pages 566–569 (cited on page 9).

[CS14] S. Chacon and B. Straub. *Pro Git*. 2nd. Berkely, CA, USA: Apress, 2014. ISBN: 1484200772, 9781484200773 (cited on pages 104, 106).

[CH08] J. D. Chai and M. Head-Gordon. "Systematic optimization of long-range corrected hybrid density functionals". In: *J. Chem. Phys.* 128.8 (2008) (cited on page 11).

[CF09] Y.-C. Cheng and G. R. Fleming. "Dynamics of Light Harvesting in Photosynthesis". In: *Annu. Rev. Phys. Chem.* 60.1 (2009), pages 241–262 (cited on pages 1, 55, 56, 58).

[CGK06] R. J. Cogdell, A. Gall, and J. Köhler. "The architecture and function of the light-harvesting apparatus of purple bacteria: from single molecules to in vivo membranes". In: *Q. Rev. Biophys.* 39.03 (2006), page 227 (cited on pages 1, 45, 50, 55–60, 63, 67, 76, 147).

[CK10] R. J. Cogdell and J. Köhler. "Sunlight, Purple Bacteria, and Quantum Mechanics: How Purple Bacteria Harness Quantum Mechanics for Efficient Light Harvesting". In: *Semicond. Semimetals*. Volume 83. C. 2010, pages 77–94 (cited on page 55).

[CMY08] A. J. Cohen, P. Mori-Sánchez, and W. Yang. "Insights into Current Limitations of Density Functional Theory". In: *Science* 321.5890 (2008), pages (cited on page 10).

[CPUa] CPU-World. *AMD Opteron 6348 specifications*. URL: http://www.cpu-world.com/CPUs/Bulldozer/AMD-Opteron%206348%20-%20OS6348WKTCGHK.html (visited on 10/21/2016) (cited on page 109).

[CPUb] CPU-World. *Intel Xeon E5-2670 specifications*. URL: http://www.cpu-world.com/CPUs/Xeon/Intel-Xeon%20E5-2670.html (visited on 10/21/2016) (cited on page 109).

[CPUc] CPU-World. *Intel Xeon E5520 specifications*. URL: http://www.cpu-world.com/CPUs/Xeon/Intel-Xeon%20E5520%20-%20AT80602002091AA%20(BX80602E5520).html (visited on 10/21/2016) (cited on page 109).

[CPUd] CPU-World. *Intel Xeon E5620 specifications*. URL: http://www.cpu-world.com/CPUs/Xeon/Intel-Xeon%20E5620%20-%20AT80614005073AB%20(BX80614E5620).html (visited on 10/21/2016) (cited on page 109).

[CM16] C. Curutchet and B. Mennucci. "Quantum Chemical Studies of Light Harvesting". In: *Chem. Rev.* (2016), acs.chemrev.5b00700 (cited on pages 1, 55, 60).

[Dah+01] M. Dahlbom, T. Pullerits, S. Mukamel, and V. Sundström. “Exciton Delocalization in the B850 Light-Harvesting Complex: Comparison of Different Measures”. In: *J. Phys. Chem. B* 105.23 (2001), pages 5515–5524 (cited on page 58).

[DR05] M. G. Dahlbom and J. R. Reimers. “Successes and failures of time-dependent density functional theory for the low-lying excited states of chlorophylls”. In: *Mol. Phys.* 103.6-8 (2005), pages 1057–1065 (cited on page 60).

[Dau+16] M. Dauth, M. Graus, I. Schelter, M. Wießner, A. Schöll, F. Reinert, and S. Kümmel. “Perpendicular Emission, Dichroism, and Energy Dependence in Angle-Resolved Photoemission: The Importance of The Final State”. In: *Phys. Rev. Lett.* 117.18 (2016), page 183001 (cited on pages 13, 26, 92).

[Dau16] M. Dauth. “Predicting photoemission: From the single-particle interpretation to the many-electron picture”. PhD thesis. University of Bayreuth, 2016 (cited on pages 13, 26, 92).

[Dav64] A. S. Davydov. “THE THEORY OF MOLECULAR EXCITONS”. In: *Sov. Phys. Uspekhi* 7.2 (1964), page 145 (cited on pages 63, 68).

[DG01] F. Della Sala and A. Görling. “Efficient localized Hartree–Fock methods as effective exact-exchange Kohn–Sham methods for molecules”. In: *J. Chem. Phys.* 115.13 (2001), page 5718 (cited on pages 9, 13).

[Dol+07] T. J. Dolinsky, P. Czodrowski, H. Li, J. E. Nielsen, J. H. Jensen, G. Klebe, and N. A. Baker. “PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations”. In: *Nucleic Acids Res.* 35.Web Server (2007), W522–W525 (cited on page 148).

[Dol+04] T. J. Dolinsky, J. E. Nielsen, J. A. McCammon, and N. A. Baker. “PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations”. In: *Nucleic Acids Res.* 32.Web Server (2004), W665–W667 (cited on page 148).

[DG90] R. Dreizler and E. Gross. *Density functional theory*. Springer-Verlag Berlin Heidelberg, 1990. ISBN: 3-540-51993-9 (cited on pages 4–6, 8).

[DH04] A. Dreuw and M. Head-Gordon. “Failure of Time-Dependent Density Functional Theory for Long-Range Charge-Transfer Excited States: The Zincbacteriochlorin-Bacteriochlorin and Bacteriochlorophyll-Spheroidene Complexes”. In: *J. Am. Chem. Soc.* 126.12 (2004), pages 4007–4016 (cited on pages 10, 50, 55, 60).

[EV93] E. Engel and S. H. Vosko. “Accurate optimized-potential-model solutions for spherical spin-polarized atoms: Evidence for limitations of the exchange-only local spin-density and generalized-gradient approximations”. In: *Phys. Rev. A* 47.4 (1993), pages 2800–2811 (cited on page 9).

[Eng+07] G. S. Engel, T. R. Calhoun, E. L. Read, T.-K. Ahn, T. Mančal, Y.-C. Cheng, R. E. Blankenship, and G. R. Fleming. “Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems”. In: *Nature* 446.7137 (2007), pages 782–786 (cited on pages 1, 55, 58).

[Fer27] E. Fermi. "Un Metodo Statistico per la determinazione di Alcune priorieta dell' atome". In: *Rend. Accad. Naz. Lincei* 6 (1927), page 602 (cited on page 3).

[Fio+03] C. Fiolhais, F. Nogueira, M. Marques, and E. Engel. "A Primer in Density Functional Theory". In: *Mater. Today* 6.12 (2003), page 59 (cited on pages 4, 8–10, 25).

[Fle+12] G. R. Fleming, G. S. Schlau-Cohen, K. Amarnath, and J. Zaks. "Design principles of photosynthetic light-harvesting". In: *Faraday Discuss.* 155 (2012), pages 27–41 (cited on pages 1, 55, 58, 59).

[Foc30] V. Fock. "Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems". In: *Zeitschrift für Phys.* 61.1-2 (1930), pages 126–148 (cited on pages 3, 8).

[För48] T. Förster. "Zwischenmolekulare Energiewanderung und Fluoreszenz". In: *Ann. Phys.* 437.1-2 (1948), pages 55–75 (cited on page 76).

[Fre+96] A. Freer, S. Prince, K. Sauer, M. Papiz, A. H. Lawless, G. McDermott, R. Cogdell, and N. W. Isaacs. "Pigment–pigment interactions and energy transfer in the antenna complex of the photosynthetic bacterium Rhodopseudomonas acidophila". In: *Structure* 4.4 (1996), pages 449–462 (cited on pages 56, 57, 147).

[FLC96] N.-U. Frigaard, K. L. Larsen, and R. P. Cox. "Spectrochromatography of photosynthetic pigments as a fingerprinting technique for microbial phototrophs". In: *FEMS Microbiol. Ecol.* 20.2 (1996), pages 69–77 (cited on page 50).

[Fri+] M. J. Frisch et al. *Gaussian 09 Revision E.01* (cited on pages 14, 43).

[GC] A. T. Gardiner and R. J. Cogdell. *Crystal structure of the LH2 complex from rhodoblastus acidophilus (strain 10050) at 1.85Å resolution.* Technical report (cited on pages 56, 147).

[Gis+99] S. J. A. van Gisbergen, P. R. T. Schipper, O. V. Gritsenko, E. J. Baerends, J. G. Snijders, B. Champagne, and B. Kirtman. "Electric Field Dependence of the Exchange-Correlation Potential in Molecular Chains". In: *Phys. Rev. Lett.* 83.4 (1999), pages 694–697 (cited on page 10).

[Git] Git. *Git.* URL: https://git-scm.com/ (visited on 10/21/2016) (cited on pages 104, 106).

[Gou61] M. Gouterman. "Spectra of porphyrins". In: *J. Mol. Spectrosc.* 6 (1961), pages 138–163 (cited on pages 45, 57, 60).

[GKG97] T. Grabo, T. Kreibich, and E. Gross. "Optimized Effective Potential for Atoms and Molecules". In: *Mol. Eng.* 7.1/2 (1997), pages 27–50 (cited on pages 9, 13).

[GG95] T. Grabo and E. Gross. "Density-functional theory using an optimized exchange-correlation potential". In: *Chem. Phys. Lett.* 240.1-3 (1995), pages (cited on pages 9, 13).

[GDP96] E. K. U. Gross, J. F. Dobson, and M. Petersilka. "Density functional theory of time-dependent phenomena". In: *Current* 181 (1996), pages 81–172 (cited on pages 6–8, 10, 43).

[GGB02] M. Grüning, O. V. Gritsenko, and E. J. Baerends. "Exchange potential from the common energy denominator approximation for the Kohn–Sham Green's function: Application to (hyper)polarizabilities of molecular chains". In: *J. Chem. Phys.* 116.15 (2002), page 6435 (cited on pages 9, 13).

[GNS07] S. Günes, H. Neugebauer, and N. S. Sariciftci. "Conjugated Polymer-Based Organic Solar Cells". In: *Chem. Rev.* 107.4 (2007), pages 1324–1338 (cited on page 1).

[HW10] G. Hager and G. Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. 1st. CRC Press, Inc., 2010. ISBN: 143981192X, 9781439811924 (cited on pages 14–17, 19, 28, 29, 36, 37, 99, 113–119, 121–123).

[Hag+10] A. Hagfeldt, G. Boschloo, L. Sun, L. Kloo, and H. Pettersson. "Dye-Sensitized Solar Cells". In: *Chem. Rev.* 110.11 (2010), pages 6595–6663 (cited on page 1).

[Han+12] M. D. M. Hanwell et al. "Avogadro: an advanced semantic chemical editor, visualization, and analysis platform." In: *J. Cheminform.* 4.1 (2012), page 17 (cited on page 58).

[Har28] D. R. Hartree. "The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods". In: *Math. Proc. Cambridge Philos. Soc.* 24.01 (1928), page 89 (cited on pages 3, 5, 8).

[Hee16] D. van Heesch. *Doxygen*. Web Resource. 2016. URL: http://www.stack.nl/~dimitri/doxygen/index.html (visited on 10/21/2016) (cited on pages 14, 104).

[HL27] W. Heitler and F. London. "Wechselwirkung neutraler Atome und homöopolare Bindung nach der Quantenmechanik". In: *Zeitschrift für Phys.* 44.6-7 (1927), pages 455–472 (cited on page 3).

[Hof+12] D. Hofmann, S. Klüpfel, P. Klüpfel, and S. Kümmel. "Using complex degrees of freedom in the Kohn-Sham self-interaction correction". In: *Phys. Rev. A* 85.6 (2012), page 062514 (cited on page 13).

[HKK10] D. Hofmann, T. Körzdörfer, and S. Kümmel. "Energy transfer and Förster's dipole coupling approximation investigated in a real-time Kohn-Sham scheme". In: *Phys. Rev. A* 82.1 (2010), page 012509 (cited on pages 13, 68, 74, 76, 78).

[HKK12] D. Hofmann, T. Körzdörfer, and S. Kümmel. "Kohn-Sham Self-Interaction Correction in Real Time". In: *Phys. Rev. Lett.* 108.14 (2012), page 146401 (cited on pages 13, 44, 45).

[HK12] D. Hofmann and S. Kümmel. "Self-interaction correction in a real-time Kohn-Sham scheme: Access to difficult excitations in time-dependent density functional theory". In: *J. Chem. Phys.* 137.6 (2012), page 064117 (cited on pages 10, 13, 44, 45).

[Hof+13] D. Hofmann-Mees, H. Appel, M. Di Ventra, and S. Kümmel. "Determining Excitation-Energy Transfer Times and Mechanisms from Stochastic Time-Dependent Density Functional Theory". In: *J. Phys. Chem. B* 117.46 (2013), pages 14408–14419 (cited on pages 65, 67, 71).

[HK64] P. Hohenberg and W. Kohn. "Inhomogeneous Electron Gas". In: *Phys. Rev.* 136.3B (1964), B864–B871 (cited on page 4).

[Hu+98] X. Hu, A. Damjanović, T. Ritz, and K. Schulten. "Architecture and mechanism of the light-harvesting apparatus of purple bacteria." In: *Proc. Natl. Acad. Sci. U. S. A.* 95.11 (1998), pages 5935–41 (cited on page 55).

[Hu+02] X. Hu, T. Ritz, A. Damjanovic, F. Autenrieth, and K. Schulten. "Photosynthetic apparatus of purple bacteria". In: *Q. Rev. Biophys.* 35.01 (2002), pages 1–62 (cited on pages 1, 45, 55–58, 76, 147).

[Inta] Intel. *Intel® Xeon® Processor E5-2670 Specifications*. URL: http://ark.intel.com/de/products/64595/Intel-Xeon-Processor-E5-2670-20M-Cache-2_60-GHz-8_00-GTs-Intel-QPI (visited on 10/21/2016) (cited on page 109).

[Intb] Intel. *Intel® Xeon® Processor E5520 Specifications*. URL: http://ark.intel.com/de/products/40200/Intel-Xeon-Processor-E5520-8M-Cache-2_26-GHz-5_86-GTs-Intel-QPI (visited on 10/21/2016) (cited on page 109).

[Intc] Intel. *Intel® Xeon® Processor E5620 Specifications*. URL: http://ark.intel.com/de/products/47925/Intel-Xeon-Processor-E5620-12M-Cache-2_40-GHz-5_86-GTs-Intel-QPI (visited on 10/21/2016) (cited on page 109).

[JAC86] D. A. H. JACOBS. "A Generalization of the Conjugate-Gradient Method to Solve Complex Systems". In: *IMA J. Numer. Anal.* 6.4 (1986), pages 447–452 (cited on page 26).

[Jor+15] J. Jornet-Somoza, J. Alberdi-Rodriguez, B. F. Milne, X. Andrade, M. A. L. Marques, F. Nogueira, M. J. T. Oliveira, J. J. P. Stewart, and A. Rubio. "Insights into colour-tuning of chlorophyll optical response in green plants". In: *Phys. Chem. Chem. Phys.* 17.40 (2015), pages 26599–26606 (cited on pages 1, 55, 60).

[JCM14] S. Jurinovich, C. Curutchet, and B. Mennucci. "The Fenna-Matthews-Olson Protein Revisited: A Fully Polarizable (TD)DFT/MM Description". In: *ChemPhysChem* 15.15 (2014), pages 3194–3204 (cited on page 76).

[Kar+11] A. Karolewski, T. Stein, R. Baer, and S. Kümmel. "Communication: Tailoring the optical gap in light-harvesting molecules". In: *J. Chem. Phys.* 134.15 (2011), page 151101 (cited on pages 9, 11, 60, 134).

[KKK13] A. Karolewski, L. Kronik, and S. Kümmel. "Using optimally tuned range separated hybrid functionals in ground-state calculations: Consequences and caveats". In: *J. Chem. Phys.* 138.20 (2013), page 204115 (cited on pages 9, 11, 60).

[KRA65] M. Kasha, H. R. Rawls, and M. Ashraf El-Bayoumi. “The exciton model in molecular spectroscopy”. In: *Pure Appl. Chem.* 11.3-4 (1965), pages 371–392 (cited on page 63).

[KB82] L. Kleinman and D. M. Bylander. “Efficacious Form for Model Pseudopotentials”. In: *Phys. Rev. Lett.* 48.20 (1982), pages 1425–1428 (cited on pages 23, 86, 131).

[Koh99] W. Kohn. “Nobel Lecture: Electronic structure of matter—wave functions and density functionals”. In: *Rev. Mod. Phys.* 71.5 (1999), pages 1253–1266 (cited on pages 3, 4).

[KS65] W. Kohn and L. J. Sham. “Self-Consistent Equations Including Exchange and Correlation Effects”. In: *Phys. Rev.* 140.4A (1965), A1133–A1138 (cited on pages 5, 8, 9, 13, 24).

[KN13] C. König and J. Neugebauer. “Exciton Coupling Mechanisms Analyzed with Subsystem TDDFT: Direct vs Pseudo Exchange Effects”. In: *J. Phys. Chem. B* 117.13 (2013), pages 3480–3487 (cited on page 57).

[KKM08] T. Körzdörfer, S. Kümmel, and M. Mundt. “Self-interaction correction and the optimized effective potential”. In: *J. Chem. Phys.* 129.1 (2008), page 014110 (cited on page 13).

[Kör+11] T. Körzdörfer, J. S. Sears, C. Sutton, and J.-L. Brédas. “Long-range corrected hybrid functionals for π-conjugated systems: Dependence of the range-separation parameter on conjugation length”. In: *J. Chem. Phys.* 135.20 (2011), page 204107 (cited on pages 9, 11, 60).

[Kre+98] T. Kreibich, S. Kurth, T. Grabo, and E. Gross. “Asymptotic Properties of the Optimized Effective Potential”. In: edited by P.-O. Löwdin. Volume 33. Advances in Quantum Chemistry. Academic Press, 1998, pages 31–48 (cited on pages 6, 10, 24, 59).

[KLI92a] J. B. Krieger, Y. Li, and G. J. Iafrate. “Construction and application of an accurate local spin-polarized Kohn-Sham potential with integer discontinuity: Exchange-only theory”. In: *Phys. Rev. A* 45.1 (1992), pages 101–126 (cited on page 9).

[KLI92b] J. B. Krieger, Y. Li, and G. J. Iafrate. “Systematic approximations to the optimized effective potential: Application to orbital-density-functional theory”. In: *Phys. Rev. A* 46.9 (1992), pages 5453–5458 (cited on page 9).

[KLI90] J. Krieger, Y. Li, and G. Iafrate. “Derivation and application of an accurate Kohn-Sham potential with integer discontinuity”. In: *Phys. Lett. A* 146.5 (1990), pages 256–260 (cited on pages 9, 13).

[Kro+06] L. Kronik, A. Makmal, M. L. Tiago, M. M. G. Alemany, M. Jain, X. Huang, Y. Saad, and J. R. Chelikowsky. “PARSEC – the pseudopotential algorithm for real-space electronic structure calculations: recent advances and novel applications to nano-structures”. In: *Phys. status solidi* 243.5 (2006), pages 1063–1079 (cited on pages 1, 13, 23, 24, 27, 38, 88, 131).

[Kro+12] L. Kronik, T. Stein, S. Refaely-Abramson, and R. Baer. "Excitation Gaps of Finite-Sized Systems from Optimally Tuned Range-Separated Hybrid Functionals". In: *J. Chem. Theory Comput.* 8.5 (2012), pages 1515–1531 (cited on pages 9, 11, 60).

[Küh95] W. Kühlbrandt. "Many wheels make light work". In: *Nature* 374.6522 (1995), pages 497–498 (cited on pages 1, 56).

[KAR01] S. Kümmel, K. Andrae, and P.-G. Reinhard. "Collectivity in the optical response. of small metal clusters". In: *Appl. Phys. B Lasers Opt.* 73.4 (2001), pages 293–297 (cited on pages 43, 45).

[KK08] S. Kümmel and L. Kronik. "Orbital-dependent density functionals: Theory and applications". In: *Rev. Mod. Phys.* 80.1 (2008), pages 3–60 (cited on pages 8–11, 13).

[KP03] S. Kümmel and J. P. Perdew. "Optimized effective potential made simple: Orbital functionals, orbital shifts, and the exact Kohn-Sham exchange potential". In: *Phys. Rev. B* 68.3 (2003), page 035103 (cited on page 6).

[Lee98] R. van Leeuwen. "Causality and Symmetry in Time-Dependent Density-Functional Theory". In: *Phys. Rev. Lett.* 80.6 (1998), pages 1280–1283 (cited on pages 6, 7).

[Lee01] R. van Leeuwen. "KEY CONCEPTS IN TIME-DEPENDENT DENSITY-FUNCTIONAL THEORY". In: *Int. J. Mod. Phys. B* 15.14 (2001), pages (cited on page 6).

[LSR02] C. Legrand, E. Suraud, and P.-G. Reinhard. "Comparison of self-interaction-corrections for metal clusters". In: *J. Phys. B At. Mol. Opt. Phys.* 35.4 (2002), pages 1115–1128 (cited on pages 44, 45).

[LSY97] R. Lehoucq, D. Sorensen, and C. Yang. *Arpack users' guide: Solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods.* 1997. URL: http://www.caam.rice.edu/software/ARPACK/UG/ug.html (visited on 12/09/2016) (cited on pages 24, 86).

[Lei+97] T. Leininger, H. Stoll, H.-J. Werner, and A. Savin. "Combining long-range configuration interaction with short-range density functionals". In: *Chem. Phys. Lett.* 275.3-4 (1997), pages 151–160 (cited on page 11).

[LPS84] M. Levy, J. P. Perdew, and V. Sahni. "Exact differential equation for the density and ionization energy of a many-particle system". In: *Phys. Rev. A* 30.5 (1984), pages 2745–2748 (cited on page 6).

[Li+98] J. Li, M. R. Nelson, C. Y. Peng, D. Bashford, and L. Noodleman. "Incorporating Protein Environments in Density Functional Theory: A Self-Consistent Reaction Field Calculation of Redox Potentials of [2Fe2S] Clusters in Ferredoxin and Phthalate Dioxygenase Reductase". In: *J. Phys. Chem. A* 102.31 (1998), pages 6311–6324 (cited on page 148).

[LKI93] Y. Li, J. B. Krieger, and G. J. Iafrate. "Self-consistent calculations of atomic properties using self-interaction-free exchange-only Kohn-Sham potentials". In: *Phys. Rev. A* 47.1 (1993), pages 165–181 (cited on page 9).

[LLS03] M. S. Liao, Y. Lu, and S. Scheiner. “Performance assessment of density-functional methods for study of charge-transfer complexes”. In: *J. Comput. Chem.* 24.5 (2003), pages 623–631 (cited on page 10).

[LKH99] J. Linnanto, J. E. I. Korppi-Tommola, and V. M. Helenius. “Electronic States, Absorption Spectrum and Circular Dichroism Spectrum of the Photosynthetic Bacterial LH2 Antenna of Rhodopseudomonas acidophila as Predicted by Exciton Theory and Semiempirical Calculations”. In: *J. Phys. Chem. B* 103.41 (1999), pages 8739–8750 (cited on page 76).

[LK06] J. Linnanto and J. Korppi-Tommola. “Quantum chemical simulation of excited states of chlorophylls, bacteriochlorophylls and their complexes”. In: *Phys. Chem. Chem. Phys.* 8.6 (2006), pages 663–687 (cited on pages 58, 60).

[LB07] E. Livshits and R. Baer. “A well-tempered density functional theory of electrons in molecules”. In: *Phys. Chem. Chem. Phys.* 9.23 (2007), pages 2932–2941 (cited on page 11).

[Mac+98] A. D. MacKerell et al. “All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins”. In: *J. Phys. Chem. B* 102.18 (1998), pages 3586–3616 (cited on pages 147, 148).

[Mar03a] M. Marques. “octopus: a first-principles tool for excited electron–ion dynamics”. In: *Comput. Phys. Commun.* 151.1 (2003), pages 60–78 (cited on pages 13, 25, 43, 44).

[MG04] M. Marques and E. Gross. “Time-Dependent Density Functional Theory”. In: *Annu. Rev. Phys. Chem.* 55.1 (2004), pages 427–455 (cited on pages 6–8, 10, 44, 45).

[Mar+12] M. a. L. Marques, N. T. Maitra, F. M. S. Nogueira, E. K. U. Gross, and A. Rubio. *Fundamentals of Time-Dependent Density Functional Theory.* Edited by M. A. Marques, N. T. Maitra, F. M. Nogueira, E. Gross, and A. Rubio. Volume 837. Lecture Notes in Physics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-23517-7 (cited on pages 6, 44, 45).

[Mar03b] R. L. Martin. “Natural transition orbitals”. In: *J. Chem. Phys.* 118.11 (2003), page 4775 (cited on pages 50, 60).

[McC95] J. D. McCalpin. “Memory Bandwidth and Machine Balance in Current High Performance Computers”. In: *IEEE Comput. Soc. Tech. Comm. Comput. Archit. Newsl.* May (1995), pages 19–25 (cited on pages 14, 30, 109).

[McD+95] G. McDermott, S. M. Prince, A. A. Freer, A. M. Hawthornthwaite-Lawless, M. Z. Papiz, R. J. Cogdell, and N. W. Isaacs. “Crystal structure of an integral membrane light-harvesting complex from photosynthetic bacteria”. In: *Nature* 374.6522 (1995), pages 517–521 (cited on pages 56, 147).

[Mei11] A. Meister. *Numerik linearer Gleichungssysteme.* 4th. Wiesbaden: Vieweg+Teubner, 2011. ISBN: 978-3-8348-1550-7 (cited on pages 14, 26, 27).

[Moo+15] B. Moore, H. Sun, N. Govind, K. Kowalski, and J. Autschbach. "Charge-Transfer Versus Charge-Transfer-Like Excitations Revisited". In: *J. Chem. Theory Comput.* 11.7 (2015), pages 3305–3320 (cited on pages 10, 60).

[Mos87] G. P. Moss. "Nomenclature of tetrapyrroles (Recommendations 1986)". In: *Pure Appl. Chem.* 59.6 (1987), pages 779–832 (cited on page 57).

[MPI12] MPI. *MPI: A Message-Passing Interface Standard. Version 3.0.* Technical report. University of Tennesse, 2012 (cited on pages 14, 16, 17, 19, 103, 104, 114).

[MUN09] M. MUNDT. "REAL-TIME APPROACH TO TIME-DEPENDENT DENSITY-FUNCTIONAL THEORY IN THE LINEAR AND NONLINEAR REGIME". In: *J. Theor. Comput. Chem.* 08.04 (2009), pages 561–574 (cited on pages 25, 26, 43–45).

[MK06] M. Mundt and S. Kümmel. "Optimized effective potential in real time: Problems and prospects in time-dependent density-functional theory". In: *Phys. Rev. A* 74.2 (2006), page 022511 (cited on pages 44, 45).

[MK07] M. Mundt and S. Kümmel. "Photoelectron spectra of anionic sodium clusters from time-dependent density-functional theory in real time". In: *Phys. Rev. B* 76.3 (2007), page 035413 (cited on pages 44, 45).

[Nee12] F. Neese. "The ORCA program system". In: *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 2.1 (2012), pages 73–78 (cited on page 148).

[Ope13] OpenMP. *OpenMP Application Program Interface - Version 4.0.* Technical report. 2013 (cited on page 17).

[ONS10] M. B. Oviedo, C. F. A. Negre, and C. G. Sánchez. "Dynamical simulation of the optical response of photosynthetic pigments". In: *Phys. Chem. Chem. Phys.* 12.25 (2010), page 6706 (cited on pages 45, 50, 57, 76, 133).

[OS11] M. B. Oviedo and C. G. Sánchez. "Transition Dipole Moments of the Q y Band in Photosynthetic Pigments". In: *J. Phys. Chem. A* 115.44 (2011), pages 12280–12285 (cited on page 57).

[Pac96] P. S. Pacheco. *Parallel Programming with MPI.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996. ISBN: 1-55860-339-5 (cited on pages 14, 16, 17).

[Pan16] D. K. Panda. *MVAPICH: MPI over InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE.* 2016. URL: http://mvapich.cse.ohio-state.edu/benchmarks/ (visited on 12/08/2016) (cited on pages 14, 113, 114).

[Pap+03] M. Z. Papiz, S. M. Prince, T. Howard, R. J. Cogdell, and N. W. Isaacs. "The Structure and Thermal Motion of the B800–850 LH2 Complex from Rps. acidophila at 2.0Å Resolution and 100K: New Structural Features and Functionally Relevant Motions". In: *J. Mol. Biol.* 326.5 (2003), pages (cited on pages 55, 56, 147).

[PZ81] J. P. Perdew and A. Zunger. "Self-interaction correction to density-functional approximations for many-electron systems". In: *Phys. Rev. B* 23.10 (1981), pages 5048–5079 (cited on pages 9, 10, 13).

[PBE96] J. P. Perdew, K. Burke, and M. Ernzerhof. “Generalized Gradient Approximation Made Simple”. In: *Phys. Rev. Lett.* 77.18 (1996), pages 3865–3868 (cited on page 9).

[PBE97] J. P. Perdew, K. Burke, and M. Ernzerhof. “Generalized Gradient Approximation Made Simple [Phys. Rev. Lett. 77, 3865 (1996)]”. In: *Phys. Rev. Lett.* 78.7 (1997), pages 1396–1396 (cited on page 9).

[PK03] J. P. Perdew and S. Kurth. “Density Functionals for Non-relativistic Coulomb Systems in the New Century”. In: *A Prim. Density Funct. Theory* 620 (2003), pages 1–55 (cited on pages 4, 8, 9).

[Per+82] J. P. Perdew, R. G. Parr, M. Levy, and J. L. Balduz. “Density-Functional Theory for Fractional Particle Number: Derivative Discontinuities of the Energy”. In: *Phys. Rev. Lett.* 49.23 (1982), pages 1691–1694 (cited on pages 6, 10).

[PW92] J. P. Perdew and Y. Wang. “Accurate and simple analytic representation of the electron-gas correlation energy”. In: *Phys. Rev. B* 45.23 (1992), pages 13244–13249 (cited on pages 9, 13).

[PGG96] M. Petersilka, U. J. Gossmann, and E. K. U. Gross. “Excitation Energies from Time-Dependent Density-Functional Theory”. In: *Phys. Rev. Lett.* 76.8 (1996), pages 1212–1215 (cited on pages 6, 43).

[Pre+92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran.* Edited by W. H. Press. 2nd edition. Cambridge University Press, 1992 (cited on pages 14, 25, 26, 44, 45, 48, 49).

[Qu+09] Z.-w. Qu, H. Zhu, V. May, and R. Schinke. “Time-Dependent Density Functional Theory Study of the Electronic Excitation Spectra of Chlorophyllide a and Pheophorbide a in Solvents”. In: *J. Phys. Chem. B* 113.14 (2009), pages 4817–4825 (cited on page 50).

[RC73] A. K. Rajagopal and J. Callaway. “Inhomogeneous Electron Gas”. In: *Phys. Rev. B* 7.5 (1973), pages 1912–1919 (cited on page 8).

[Rog+12] H.-H. Rogner, R. F. Aguilera, C. Archer, R. Bertani, S. C. Bhattacharya, M. B. Dusseault, L. Gagnon, H. Haberl, M. Hoogwijk, A. Johnson, M. L. Rogner, H. Wagner, and V. Yakushev. “Chapter 7 - Energy Resources and Potentials”. In: *Glob. Energy Assess. - Towar. a Sustain. Futur.* Edited by J. Zou. Cambridge University Press, Cambridge, UK, New York, NY, USA, and the International Institute for Applied Systems Analysis, Laxenburg, Austria, 2012, pages 423–512 (cited on page 1).

[RG84] E. Runge and E. K. U. Gross. “Density-Functional Theory for Time-Dependent Systems”. In: *Phys. Rev. Lett.* 52.12 (1984), pages 997–1000 (cited on pages 6, 7, 13).

[SS96] U. Saalmann and R. Schmidt. “Non-adiabatic quantum molecular dynamics: basic formalism and case study”. In: *Zeitschrift für Phys. D Atoms, Mol. Clust.* 38.2 (1996), pages 153–163 (cited on pages 1, 25, 43).

[SFK09] E. Sagvolden, F. Furche, and A. Köhn. “Förster Energy Transfer and Davydov Splittings in Time-Dependent Density Functional Theory: Lessons from 2-Pyridone Dimer”. In: *J. Chem. Theory Comput.* 5.4 (2009), pages (cited on page 76).

[SF95] A. Savin and H.-J. Flad. “Density functionals for the Yukawa electron-electron interaction”. In: *Int. J. Quantum Chem.* 56.4 (1995), pages 327–332 (cited on page 11).

[Sch16] P. Schaffhauser. “Ladungstransport in molekularen Systemen und Visualisierung molekularer Strukturen”. PhD thesis. University of Bayreuth, 2016 (cited on pages 13, 26, 92).

[SK16] P. Schaffhauser and S. Kümmel. “Using time-dependent density functional theory in real time for calculating electronic transport”. In: *Phys. Rev. B* 93.3 (2016), page 35115 (cited on pages 13, 26).

[Sch13] I. Schelter. “Numerische Lösung des Optimized Effective Potential”. Master’s thesis. University of Bayreuth, 2013 (cited on page 9).

[SR06] G. D. Scholes and G. Rumbles. “Excitons in nanoscale systems”. In: *Nat. Mater.* 5.9 (2006), pages 683–696 (cited on pages 55, 58, 63, 76).

[Sch15] Schrödinger, LLC. “The PyMOL Molecular Graphics System, Version 1.8”. 2015 (cited on page 57).

[Sch26] E. Schrödinger. “Quantisierung als Eigenwertproblem - Erste Mitteilung”. In: *Ann. Phys.* 79 (1926), pages 361–376 (cited on page 3).

[Sei+96] A. Seidl, A. Görling, P. Vogl, J. A. Majewski, and M. Levy. “Generalized Kohn-Sham schemes and the band-gap problem”. In: *Phys. Rev. B* 53.7 (1996), pages 3764–3774 (cited on pages 9, 12).

[Sha+15] Y. Shao et al. “Advances in molecular quantum chemistry contained in the Q-Chem 4 program package”. In: *Mol. Phys.* 113.2 (2015), pages 184–215 (cited on pages 43, 60, 61, 131).

[SH53] R. T. Sharp and G. K. Horton. “A Variational Approach to the Unipotential Many-Electron Problem”. In: *Phys. Rev.* 90.2 (1953), pages 317–317 (cited on page 9).

[SS07] V. Simoncini and D. B. Szyld. “Recent computational developments in Krylov subspace methods for linear systems”. In: *Numer. Linear Algebr. with Appl.* 14.1 (2007), pages 1–59 (cited on pages 26, 27).

[Sol16] C. Soldner. “Untersuchung des Energietransfers in molekularen Systemen”. Master’s thesis. University of Bayreuth, 2016 (cited on page 73).

[SKB09a] T. Stein, L. Kronik, and R. Baer. “Prediction of charge-transfer excitations in coumarin-based dyes using a range-separated functional tuned from first principles”. In: *J. Chem. Phys.* 131.24 (2009), page 244119 (cited on pages 9, 11, 60, 134).

[SKB09b] T. Stein, L. Kronik, and R. Baer. “Reliable Prediction of Charge Transfer Excitations in Molecular Complexes Using Time-Dependent Density Functional Theory”. In: *J. Am. Chem. Soc.* 131.8 (2009), pages 2818–2820 (cited on pages 9–11, 60, 134).

[Ste+94] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch. "Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields". In: *J. Phys. Chem.* 98.45 (1994), pages 11623–11627 (cited on pages 9, 11, 148).

[Sti16] A. Stiller. *Intel bringt (endlich) den HPC-Prozessor Xeon Phi Knights Landing*. 2016. URL: http://www.heise.de/newsticker/meldung/Intel-bringt-endlich-den-HPC-Prozessor-Xeon-Phi-Knights-Landing-3241370.html (visited on 11/01/2016) (cited on page 16).

[SŞS12] J. Strümpfer, M. Şener, and K. Schulten. "How Quantum Coherence Assists Photosynthetic Light-Harvesting". In: *J. Phys. Chem. Lett.* 3.4 (2012), pages 536–542 (cited on pages 1, 55, 58).

[Sun03] D. Sundholm. "A density-functional-theory study of bacteriochlorophyll b". In: *Phys. Chem. Chem. Phys.* 5.19 (2003), page 4265 (cited on page 50).

[Sun00] D. Sundholm. "Comparison of the electronic excitation spectra of chlorophyll a and pheophytin a calculated at density functional theory level". In: *Chem. Phys. Lett.* 317.6 (2000), pages 545–552 (cited on pages 50, 57, 60).

[Sun99] D. Sundholm. "Density functional theory calculations of the visible spectrum of chlorophyll a". In: *Chem. Phys. Lett.* 302.5-6 (1999), pages 480–484 (cited on pages 50, 60).

[Swi15] N. Swiridoff. "Untersuchung des Einflusses von Effektivpotentialen auf die elektronische Struktur von Bacteriochlorophyll". Bachelor's thesis. University of Bayreuth, 2015 (cited on page 59).

[TS76] J. D. Talman and W. F. Shadwick. "Optimized effective atomic central potential". In: *Phys. Rev. A* 14.1 (1976), pages 36–40 (cited on page 9).

[TK09] M. Thiele and S. Kümmel. "Photoabsorption spectra from adiabatically exact time-dependent density-functional theory in real time". In: *Phys. Chem. Chem. Phys.* 11.22 (2009), page 4631 (cited on pages 43–45).

[Tho27] L. H. Thomas. "The calculation of atomic fields". In: *Math. Proc. Cambridge Philos. Soc.* 23.05 (1927), page 542 (cited on page 3).

[TF08] B. C. Thompson and J. M. J. Fréchet. "Polymer–Fullerene Composite Solar Cells". In: *Angew. Chemie Int. Ed.* 47.1 (2008), pages 58–77 (cited on page 1).

[TEE10] B. Tian, E. S. E. Eriksson, and L. A. Eriksson. "Can Range-Separated and Hybrid DFT Functionals Predict Low-Lying Excitations? A Tookad Case Study". In: *J. Chem. Theory Comput.* 6.7 (2010), pages 2086–2094 (cited on page 60).

[Toz03] D. J. Tozer. "Relationship between long-range charge-transfer excitation energy error and integer discontinuity in Kohn-Sham theory". In: *J. Chem. Phys.* 119.24 (2003) (cited on pages 10, 50, 55, 60).

[THW10] J. Treibig, G. Hager, and G. Wellein. "Likwid: A lightweight performance-oriented tool suite for x86 multicore environments". In: *2010 39th Int. Conf. Parallel Process. Work.* IEEE. 2010, pages 207–216 (cited on pages 14, 122).

[TM91] N. Troullier and J. L. Martins. "Efficient pseudopotentials for plane-wave calculations". In: *Phys. Rev. B* 43.3 (1991), pages 1993–2006 (cited on pages 23, 86, 131).

[Tur+00] W. C. Turkenburg, J. Beurskens, A. Faaij, P. Fraenkel, I. Fridleifsson, E. Lysen, D. Mills, J. R. Moreira, L. J. Nilsson, A. Schaap, and W. C. Sinke. *Chapter 7 - Renewable Energy Technologies.* United Nations Development Programme et al., 2000 (cited on page 1).

[Tur+12] W. C. Turkenburg, D. J. Arent, R. Bertani, A. Faaij, M. Hand, W. Krewitt, E. D. Larson, J. Lund, M. Mehos, T. Merrigan, C. Mitchell, J. R. Moreira, W. Sinke, V. Sonntag-O'Brien, B. Thresher, W. van Sark, E. Usher, and E. Usher. "Chapter 11 - Renewable Energy". In: *Glob. Energy Assess. - Towar. a Sustain. Futur.* Cambridge University Press, Cambridge, UK, New York, NY, USA, and the International Institute for Applied Systems Analysis, Laxenburg, Austria, 2012, pages 761–900 (cited on page 1).

[UB14] G. M. Ullmann and E. Bombarda. "Continuum Electrostatic Analysis of Proteins". In: *Protein Model.* Edited by G. Náray-Szabó. Cham: Springer International Publishing, 2014, pages 135–163 (cited on pages 59, 147, 148).

[UU12] R. T. Ullmann and G. M. Ullmann. "GMCT : A Monte Carlo simulation package for macromolecular receptors". In: *J. Comput. Chem.* 33.8 (2012), pages 887–900 (cited on page 147).

[VÖC99] I. Vasiliev, S. Öğüt, and J. R. Chelikowsky. "Ab Initio Excitation Spectra and Collective Electronic Response in Atoms and Clusters". In: *Phys. Rev. Lett.* 82.9 (1999), pages 1919–1922 (cited on pages 50, 133).

[Veg+15] C. P. van der Vegte, J. D. Prajapati, U. Kleinekathöfer, J. Knoester, and T. L. C. Jansen. "Atomistic Modeling of Two-Dimensional Electronic Spectra and Excited-State Dynamics for a Light Harvesting 2 Complex". In: *J. Phys. Chem. B* 119.4 (2015), pages 1302–1313 (cited on pages 58, 59).

[VB07] Z. Vokáčová and J. V. Burda. "Computational Study on Spectral Properties of the Selected Pigments from Various Photosystems: Structure-Transition Energy Relationship". In: *J. Phys. Chem. A* 111.26 (2007), pages 5864–5878 (cited on page 57).

[VS06] O. A. Vydrov and G. E. Scuseria. "Assessment of a long-range corrected hybrid functional". In: *J. Chem. Phys.* 125.23 (2006) (cited on pages 9, 11, 12).

[WU08] H. O. Wijewardane and C. A. Ullrich. "Real-Time Electron Dynamics with Exact-Exchange Time-Dependent Density-Functional Theory". In: *Phys. Rev. Lett.* 100.5 (2008), page 056404 (cited on pages 9, 44, 45).

[YB98] K. Yabana and G. F. Bertsch. "Oscillator strengths with pseudopotentials". In: *Phys. Rev. A* 58.3 (1998), pages 2604–2607 (cited on page 44).

[YB96] K. Yabana and G. F. Bertsch. "Time-dependent local-density approximation in real time". In: *Phys. Rev. B* 54.7 (1996), pages 4484–4487 (cited on pages 1, 25, 43–45).

[YB99] K. Yabana and G. F. Bertsch. "Time-dependent local-density approximation in real time: Application to conjugated molecules". In: *Int. J. Quantum Chem.* 75.1 (1999), pages 55–66 (cited on pages 1, 25, 43–45).

[YTH04] T. Yanai, D. P. Tew, and N. C. Handy. "A new hybrid exchange–correlation functional using the Coulomb-attenuating method (CAM-B3LYP)". In: *Chem. Phys. Lett.* 393.1-3 (2004), pages 51–57 (cited on page 60).

[ZS80] A. Zangwill and P. Soven. "Density-functional approach to local-field effects in finite systems: Photoabsorption in the rare gases". In: *Phys. Rev. A* 21.5 (1980), pages 1561–1572 (cited on pages 6, 44).

Danksagung

Im Laufe der letzten Jahre während meiner Studienzeit in Bayreuth habe ich viele Bekanntschaften geschlossen und gute Freunde gefunden, die mich auf dieser aufregenden Etappe meines Lebens begleitet haben. Genauso gibt es etliche Personen, die ich schon viel länger kenne und die immer an mir festgehalten haben. Ich möchte diese Zeilen nutzen, um mich bei ihnen zu bedanken.

Allen voran danke ich Stephan Kümmel, der mich während meines Studiums und schließlich als Doktorvater über viele Jahre hinweg betreut hat. Neben seinen fachlichen Fähigkeiten haben mich besonders seine menschliche Art und die Begeisterung für die Physik, mit der er mich immer wieder mitreißen konnte, fasziniert. In diesen Jahren war Stephan nicht nur ein herausragender Mentor sondern auch ein sehr guter Kollege und Freund.

Genauso danke ich den Freunden, die ich im Laufe der Zeit am Lehrstuhl "Theoretische Physik IV" gefunden habe. Darunter sind Linn Leppert, Dirk Hofmann-Mees, Andreas Karolewski, Tobias Schmidt, Matthias Dauth, Thiago de Queiroz, Philipp Schaffhauser, Thilo Aschebrock, Fabian Hofmann, Julian Garhammer, Christoph Soldner und Alexander Kaiser. Besonders herausheben möchte ich Dirk, Thiago, Christoph und Alexander, die mit mir zeitweise dasselbe Büro geteilt haben, sowie Thiago, Philipp und Matthias für die großartige Zusammenarbeit bei verschiedenen Projekten.

Über meinen Lehrstuhl hinaus danke ich weiterhin dem Graduiertenkolleg GRK1640, insbesondere meinen Mentoren Matthias Ullmann und Richard Hildner sowie Johannes Förster für die sehr gute Zusammenarbeit und Freundschaft.

Natürlich würden alle Rädchen stillstehen ohne die guten Seelen, die sich unermüdlich um allerhand Organisatorisches und die Administration der Computer kümmern und bei jeder Gelegenheit zur Stelle sind. Für ihre Hilfsbereitschaft und ihr Engagement bedanke ich mich ganz herzlich bei Monika Birkelbach, Claudia Geier, Markus Hilt, Bernhard Winkler und Werner Reichstein.

An dieser Stelle danke ich auch allen Freunden, die ich während der Schulzeit, des Studiums und meiner Zeit in der Fachschaftsvertretung und im Absolventenverein meiner Fakultät kennengelernt habe. Insbesondere Lisa Wohlrab und Dominic Raithel sind mir seit dem ersten Semester sehr gute Freunde, auf die ich mich immer verlassen kann. Noch weiter zurück reicht die Freundschaft zu Jonas Zeitler, Florian Wilhelm und Christopher Michael, die mich schon seit vielen Jahren durch dick und dünn begleiten.

Besonders danke ich natürlich meiner Familie, die mir auf meinem Weg von Beginn an alles ermöglicht und mich immer unterstützt hat. Darunter sind meine Eltern Ulrike und Jürgen Schelter, meine Großeltern Hanna und Fritz Schelter und Gisela und Georg Macht, mein Onkel Harald Schelter und meine Paten Waltraud und Werner Heußinger. Ebenso danke ich der Familie meiner Freundin, insbesondere Roswitha und Roland Kopp und Marianne Seyferth, dafür, dass sie mich ganz herzlich in ihre Mitte aufgenommen haben. Am Ende gilt mein ganz besonderer Dank meiner Freundin Eva, auf die ich mich immer verlassen kann und die stets zu mir gehalten und mich in schwierigen Zeiten unterstützt hat.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin erkläre ich, dass ich die Hilfe von gewerblichen Promotionsberatern bzw. -vermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen habe, noch künftig in Anspruch nehmen werde.

Zusätzlich erkläre ich hiermit, dass ich keinerlei frühere Promotionsversuche unternommen habe.

Ort, Datum	Unterschrift (Ingo Schelter)