INAUGURAL - DISSERTATION

zur Erlangung der Doktorwürde der
Naturwissenschaftlich-Mathematischen Gesamtfakultät der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von

Maximilian Scheurer, M.Sc.

aus Kördorf

Datum der mündlichen Prüfung: 27. Oktober 2021

Quantum Chemical Methods for Molecular Properties in Complex Environments

Maximilian Scheurer

im August 2021

Gutachter:
Prof. Dr. Andreas Dreuw
Prof. Dr. Oriol Vendrell

Für Mama, Papa und Lukas

Zusammenfassung

Quantenchemische Simulationen molekularer Eigenschaften sind essentiell, um einen detaillierten Einblick in eine Vielzahl chemischer und biologischer Prozesse zu erhalten. Insbesondere bei der Untersuchung lichtinduzierter Reaktionen ist die Interpretation experimenteller Ergebnisse ohne begleitende computerbasierte Modellierung der zugrunde liegenden elektronischen Anregungen kaum denkbar. Durch die Komplexität der beteiligten molekularen Systeme in Bezug auf elektronische Struktur, intermolekulare Wechselwirkungen und Dynamik stößt man jedoch an die Grenzen der verfügbaren Rechenkapazität. Abhilfe können hybride quanten-klassische Umgebungsschemata leisten, die die gesamte Komplexität durch Aufspaltung des Systems in eine Quantenregion und ihre Umgebung erfassen, was zu einer erheblichen Verringerung der benötigten Rechenressourcen führt. Diese Schemata behalten somit die quantenchemische Beschreibung für den relevantesten Teil bei, ohne den Einfluss der Umgebung zu vernachlässigen.

In dieser Dissertation entwickle ich Methoden zur Modellierung molekularer Eigenschaften in komplexen Umgebungen. Der erste Teil der Arbeit widmet sich kombinierten Schemata aus dem *polarizable embedding* (PE)-Modell und dem algebraisch-diagrammatischen Konstruktionsverfahren (ADC) des Polarisationspropagators für die Simulation spektroskopischer Eigenschaften. Ich leite insgesamt zwei solcher Schemata her: Das erste Schema – pt-PE-ADC – verwendet einen selbstkonsistenten PE-Referenzzustand mit einem kanonischen ADC-Verfahren und eignet sich zur Berechnung elektronischer Anregungsenergien mittels störungstheoretischer Korrekturen. Das zweite Schema – LR-PE-ADC – beinhaltet eine direkte iterative Kopplung an die polarisierbare Umgebung, wodurch es sowohl für angeregte elektronische Zustände als auch für Response-Eigenschaften höherer Ordnung geeignet ist. Darüber hinaus leite ich Gleichungen für analytische Kerngradienten mit PE-ADC her. Um die Verfügbarkeit des PE-Modells im Allgemeinen zu verbessern, implementiere ich eine eigenständige, quelloffene und hybride Python/C++-Bibliothek namens CPPE, die bereits in mehreren quantenchemischen Programmen zur Verfügung steht. Die PE-ADC-Schemata werden in `adcc`, einem Toolkit zur Entwicklung von ADC-basierten Methoden, welches mit mehreren Python-basierten Quantenchemieprogrammen kombinierbar ist, implementiert. Das schlichte Design beider Bibliotheken ermöglicht die

Erweiterung bestehender Workflows und *rapid prototyping*. Außerdem implementiere ich Response-Eigenschaften für ADC mithilfe der *intermediate state representation* (ISR) in einer neuen Python-Bibliothek namens `respondo`. Das Zusammenwirken dieser drei Bibliotheken ermöglicht es dem Anwender, neue Funktionen einfach zu implementieren, während Benutzerfreundlichkeit und Effizienz für praktische Berechnungen erhalten bleiben. Die verschiedenen Schemata teste ich in mehreren Benchmark-Rechnungen und Fallstudien. Beispielsweise zeige ich, dass die Fehler der Anregungsenergien bei Verwendung von pt-PE-ADC für solvatisiertes *p*-Nitroanilin wesentlich kleiner sind als die intrinsischen Fehler des ADC-Verfahrens. Außerdem untersuche ich den *charge transfer* (CT)-Zustand, der beim Mechanismus der Photoprotektion im Flavoprotein Dodecin beteiligt ist. Ich führe zudem die ersten Berechnungen von Response-Eigenschaften höherer Ordnung mit ADC und einem polarisierbaren Modell durch. In diesen Studien beobachte ich, dass LR-PE-ADC die Genauigkeit der berechneten Eigenschaften im Vergleich zu einfacheren Kopplungsschemata stark verbessert. Weiterhin zeige ich, dass Korrekturen für Artefakte des PE-Modells und die physikalisch korrekte Beschreibung der PE-ADC-Intensitäten bei Benchmarks mit Supersystem-Berechnungen entscheidend sind. Mit meinen theoretischen Herleitungen und frei verfügbaren Implementierungen stelle ich meines Wissens den bisher vollständigsten Funktionsumfang polarisierbarer Modelle in Kombination mit ADC zur Verfügung.

Der zweite Teil der Arbeit beinhaltet zunächst eine allgemeine Verbesserung der Performance von PE. Ich implementiere ein PE-Schema, bei dem die elektrischen Felder in der Umgebung mit der *fast-multipole*-Methode (FMM) statt durch direkte Summation ausgewertet werden. Folglich skaliert die Berechnung der elektrischen Felder als geschwindigkeitsbestimmender Schritt des klassischen Teils in PE-FMM asymptotisch linear, sodass effiziente Simulationen von Umgebungen mit über einer Million polarisierbarer Atome möglich sind. Des Weiteren zeige ich algorithmische Details für die numerisch stabile Lösung von Response-Gleichungen im ADC/ISR-Framework und analysiere das Konvergenzverhalten verschiedener Lösungsalgorithmen. Diese Algorithmen sind auch für die effiziente Lösung von PE-ADC-Response-Gleichungen vorteilhaft. Ich präsentiere außerdem Herleitungen und numerische Studien komplexer Polarisierbarkeiten für elektronisch angeregte Zustände, womit das ADC/ISR-Framework über Response-Eigenschaften des Grundzustands hinaus erweitert wird. Dann untersuche ich die Verzerrung von Molekülen durch äußere Kräfte. Ich entwickle eine neue Methode zur Anwendung von hydrostatischem Druck in quantenchemischen Simulationen durch Gauß-Potentiale. Dieses implizite Umgebungsschema namens GOSTSHYP übt durch Kompression der Elektronendichte Druck auf ein Molekül oder Atom aus, sodass es möglich wird, Geometrieoptimierungen und Dynamiksimulationen bei einem festgelegten Druck durchzuführen. Dieser Funktionsumfang ist in keiner anderen vergleichbaren Methode zu finden. Ich verwende Molekulardynamiksimulationen mit quantenchemischer Kraftanalyse zur Aufklärung des Entfaltungsprozesses von Rubredoxin, dessen aktives Zentrum bereits durch sehr geringe Krafteinwirkung gebrochen wird. Mit meinen Berechnungen zeige ich, dass dieses Phänomen nicht – wie bisher in der Literatur angenommen – aus Wasserstoffbrückenbindungen zum Protein resultiert. Schließlich präsentiere ich das Design neuartiger photolabiler Schutzgruppen auf Basis von Fluorenderivaten. Unter Verwendung eines effizienten computergestützten Screening-Protokolls schlage ich Cyclopentadithiophen als molekulares Gerüst vor, was zur nächsten Generation fluorenbasierter photolabiler Schutzgruppen mit verbesserten Absorptions- und Reaktionseigenschaften führt.

Abstract

Quantum chemical simulations of molecular properties are crucial to obtain in-depth insight into a multitude of chemical and biological phenomena. In particular for investigating light-driven systems, modeling of electronic excitations by computational means is indispensable for supporting, complementing, and extending experimental findings. The complexity in terms of electronic structure, intermolecular interactions, and dynamics of the involved molecular systems, however, pushes the limits of computational feasibility. Hybrid quantum-classical environment schemes tackle this complexity by splitting the system into a quantum region and its environment. Thus, they retain the high-level quantum chemical description for the part of interest without neglecting the pivotal effects of the environment.

In this thesis, I develop methods for modeling molecular properties in complex environments. The first half of the thesis is dedicated to new combined approaches of the polarizable embedding (PE) model and the algebraic-diagrammatic construction (ADC) scheme for the polarization propagator for computational spectroscopy simulations. I derive and implement two PE-ADC coupling schemes: The first scheme – pt-PE-ADC – uses a self-consistent PE reference state with a canonical ADC procedure and is suited for computation of electronic excitation energies including *a posteriori* perturbative corrections. The second scheme – LR-PE-ADC – includes direct coupling to the polarizable environment in an iterative manner, making it suitable for excited electronic states and higher-order response properties. Furthermore, I derive working equations to evaluate analytic nuclear gradients using PE-ADC. To advance the availability of the PE model in general, I implement a standalone, open-source, and hybrid Python/C++ library, called CPPE, and interface it to several freely available quantum chemical host programs. The PE-ADC schemes are implemented with `adcc`, a toolkit for development of ADC-based methods and combinable with several Python-driven host programs. The simple and clean design of both libraries allows for extension of existing workflows and rapid prototyping. Moreover, I implement response properties using ADC and the intermediate state representation (ISR) in a new Python library, called `respondo`. The synergy of all three libraries enables the user to implement new features in a straightforward manner, while maintaining usability and efficiency for practical calculations. I test the individual approaches in several benchmark calculations and case studies. For example, I find that excitation energy errors using pt-PE-ADC for microsolvated *p*-nitroaniline are much smaller than the intrinsic error of ADC itself. Furthermore, I investigate the charge transfer (CT) state involved in the photoprotection mechanism of the flavoprotein dodecin. In addition, I conduct the first computations of higher-order response properties with ADC and a polarizable model. In these studies, I observe that LR-PE-ADC greatly improves the accuracy of the property compared to simpler coupling schemes. I further show that corrections for electron spill-out artifacts and the physically sound evaluation of PE-ADC intensities are decisive when benchmarking against supersystem calculations. With my theoretical derivations and open-source implementations, I provide, to the best of my knowledge, the most complete and unique feature set of polarizable models combined with ADC to date.

The second half of the thesis first contains a general performance improvement of PE models. I implement a PE scheme where the classical electric fields in the environment are evaluated using the fast multipole method (FMM) instead of direct summation. Consequently, the electric field evaluations as rate-limiting step of the classical part exhibit an asymptotic linear scaling in the PE-FMM scheme, making it suitable for efficient simulations of environments with over a million polarizable sites. Next, I show algorithm details for numerically stable solution of response equations in the ADC/ISR framework, and I analyze convergence behavior of different solver algorithms. These algorithms are beneficial for efficient evaluation of PE-ADC response properties, too. I present derivations and numerical case studies of complex excited state polarizabilities which extend the ADC/ISR framework beyond ground state response properties. Then, I investigate the distortion of molecules under external forces. I develop a new electronic structure method to apply hydrostatic pressure in standard quantum chemical simulations via Gaussian potentials, called GOSTSHYP. This implicit embedding scheme directly exerts pressure on a molecule via compression of the electron density, such that it becomes possible to treat atoms and molecules and to run geometry optimizations and dynamics simulations at a pre-defined pressure. This feature set is not found in any other comparable method. I use steered molecular dynamics (SMD) simulations with quantum chemical strain analysis tools to elucidate the rupture process of rubredoxin. I prove that the extremely low rupture force does not result from hydrogen bond networks to the protein as assumed so far in the literature, but that its origin is likely more intricate. Finally, I present the design of novel photocages based on fluorene derivatives. Using an efficient computational screening protocol, I propose cyclopenta-dithiophene as scaffold, leading to the next generation of fluorene-based photocages with desirable absorption and uncaging properties.

Contents

Abbreviations and Acronyms

***p*NA**	*p*-nitroaniline
ADC	algebraic-diagrammatic construction
AIMD	*ab initio* molecular dynamics
AO	atomic orbital
API	application programming interface
BLG	Beta-lactoglobulin
BOMD	Born-Oppenheimer molecular dynamics
CC	coupled cluster
CG	conjugate gradient
CPP	complex polarization propagator
CPPE	C++ and Python library for polarizable embedding
CSE	common subexpression elimination
CT	charge transfer
DFT	density functional theory
ECP	effective core potential
EEF	effective external fields
EFP	effective fragment potential
EOM	equation of motion
ERI	electron-repulsion integral
FA	formaldehyde
FMM	fast multipole method
GOSTSHYP	Gaussians on surface tesserae simulate hydrostatic pressure
HF	Hartree-Fock
IS	intermediate state
ISR	intermediate state representation
LCAO	linear combination of atomic orbitals
LE	local excitation
Lf	lumiflavin
LR	linear response

MD	molecular dynamics
MECP	minimum energy crossing point
MM	molecular mechanics
MO	molecular orbital
MP	Møller-Plesset
ON	occupation number
PCG	preconditioned conjugate gradient
PCM	polarizable continuum model
PE	polarizable embedding
pt	perturbative
PTE	perturbation in the energy
RHF	restricted Hartree-Fock
RI	resolution of the identity
SCF	self-consistent field
SMD	steered molecular dynamics
SOPPA	second-order polarization propagator approximation
SOS	sum-over-states
SQ	second quantization
STL	standard template library
TDA	Tamm-Dancoff approximation
TDDFT	time-dependent density functional theory
TP	two-photon
TPA	two-photon absorption
TSSF	tessellation sphere scaling factor
UHF	unrestricted Hartree-Fock

Chapter 1

Introduction

Accurate modeling of molecular properties is the cornerstone of modern quantum and computational chemistry. The vast availability of electronic structure methods together with principles to obtain almost any kind of molecular property allows for the routine application of computational simulations to complement and understand experimental observations.[1–3] Such applications include, e.g., the simulation of various spectroscopic properties. Electronic structure methods enable the computational investigation of ever more complex systems – however, they are not the only ingredient required. Due to the computational cost of *ab initio* quantum chemical methods, the size of the system under study is limited, and often, these high-level methods cannot capture its entirety. Fortunately, a smaller subsystem where a certain process occurs can usually be extracted. For example, this could be the active site of a photoreactive protein system where the electronic excitation process is primarily localized. Then the smaller subsystem can still be treated using high-level *ab initio* methods, but interactions with the rest of the system are ignored. This simplification must be made carefully to balance the trade-off between computational feasibility on the one hand, and an accurate description of the system on the other hand: Since most chemical processes do not take place in vacuum but in the condensed phase, the influences of the environment are typically non-negligible, and might even be pivotal to accurately discern the property of interest. To retain exactly these non-negligible interactions with the molecular environment, a plethora of environment models

---

Parts of this chapter have already been published in:

- M. Scheurer, M. F. Herbst, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted, "Polarizable Embedding Combined with the Algebraic Diagrammatic Construction: Tackling Excited States in Biomolecular Systems", *J. Chem. Theory Comput.* **2018**, *14 (9)*, 4870-4883.
- M. Scheurer, "Polarizable Embedding for the Algebraic-Diagrammatic Construction Scheme", *Springer Fachmedien Wiesbaden*, **2020**.

exist.[4–8] Each environment model captures important physical properties in a simplified manner tailored to the requirements of the scientific question at hand, e.g., through a lower-level quantum chemical method, or through purely classical electrostatic interactions.[4,9] These methods are referred to as hybrid quantum-quantum and quantum-classical embedding schemes. Two main classes of hybrid quantum-classical environment schemes – implicit and explicit embedding schemes – are commonly employed. Implicit embedding schemes mimic the medium surrounding a specific molecule by a dielectric continuum. For these continuum solvation models (CSM), there are multiple formulations, e.g., the polarizable continuum model (PCM) and the conductor-like screening model (COSMO).[10–12] Calculations involving CSM enable the description of a liquid solvent, and are straightforward to perform since only a dielectric constant and the molecular cavity need to be chosen. One of the most severe shortcomings of CSM is the missing directionality of explicit solvent interactions.[12,13] These interactions are, however, crucial in anisotropic, heterogeneous environments, such as protic solvents, proteins, nucleic acids, and biological membranes. Explicit embedding schemes capture this heterogeneity by retaining a fully atomistic environment.[5] As a result, explicit quantum-classical embedding schemes are generally more complex and computationally demanding than CSM as they require parametrization of the atomistic environment. In the simplest case, these parameters can be obtained from molecular mechanics (MM) force fields which are rarely constructed to reproduce the microscopic detail required for accurate embedding calculations, especially with respect to molecular response properties for computational spectroscopy. In fact, a description of permanent electrostatics *and* mutual polarization effects between the environment and the quantum region is important. To this end, polarizable embedding (PE) models take into account electrostatic and mutual polarization interactions between the quantum region and the atomistic environment.[7,8,14–22] Due to their reliability, accuracy, and versatility, PE models are becoming increasingly popular for a variety of applications, e.g., to address light-driven biochemical systems with respect to both dynamics and spectroscopy.[8,20–22] Quantum-classical PE models are considered the “future gold standard” technique for simulation of complex systems.[20] They model electrostatic interactions of the environment with the quantum region through a multi-center multipole expansion together with atomic polarizabilities located at the expansion sites. This formulation entails a fully self-consistent treatment of polarization effects between the environment and the quantum region, suitable for the description of the electronic ground state as well as excited states.[14] While different flavors of hybrid PE schemes exist, the overall formulation can be generalized and expressed in a coherent manner.[21,23] In addition to the choice of the embedding scheme, a reliable electronic structure method for the quantum region is a decisive factor for the quality of the computed molecular property. Currently, the most popular method is density functional theory (DFT), which has severe shortcomings compared to more expensive, *ab initio* wave function based methods.[24–26] Since DFT-based methodologies are not systematically improvable, they can only in rare cases be used as black-box methods and must otherwise be carefully benchmarked against *ab initio* references. The use of wave function methods in combination with atomistic polarizable models is still not common practice yet, even though some combinations using coupled cluster (CC),[27–29] the second-order polarization propagator approximation (SOPPA),[30] and the algebraic-diagrammatic construction (ADC) scheme for the polarization propagator exist.[31–33] Especially the ADC scheme is popular for its beneficial properties and its computational efficiency.[31,34–36] The ADC scheme for the polarization propagator can be used for simulating excited electronic states and for properties of higher order.[34–37] By this versatility, combination of ADC with a PE model makes it possible to compute an extensive set of molecular properties in atomistic polarizable environments.

For PE models to become the future gold standard for modeling molecular properties, they must be freely available to a broad range of potential users through open-source computational toolkits. Through community development endeavors, an integrative environment is created, and quantum chemical method development becomes more sustainable, more accessible, and independently verifiable.[38] Two prominent examples for open-source quantum chemistry toolkits are Psi4 and PySCF,[38,39] each providing a rich and competitive feature set of electronic structure methods. Everyone can contribute features and bug fixes for these toolkits via GitHub, a collaborative developer platform, and these contributions are carefully reviewed by the maintainers and other developers before integrating them into the main branch of the repository. Furthermore, these toolkits rely on program and library interoperability to carry out specific tasks, such that their code components become disentangled and are often maintained in third-party repositories, e.g., computation of molecular integrals, inclusion of environment models, or post-processing routines. Taking advantage of a rich ecosystem of different programs and libraries pushes the frontiers of today's quantum chemistry simulation workflows. It is important to consider open-source development for polarizable models which can only become the gold standard technique in the future if they are easily accessible in many (free) programs combined with various quantum chemical methods.

In this thesis, I develop and implement quantum chemical methods for the description of molecular properties in different complex environments. I provide a comprehensive suite of combined methodologies using the ADC scheme for the polarization propagator and the PE model, called PE-ADC approaches in the following. Relying on the PE formulation by Kongsted and co-workers[14,15] together with ADC as *ab initio* wave function method of choice, I present hybrid quantum-classical approaches for obtaining excited states, higher-order response properties, and analytic nuclear gradients. This entails a rigorous theoretical derivation of all working equations, together with a fully open-source implementation of the relevant methodologies. Moreover, I verify, benchmark, and illustrate the combined PE-ADC approaches in several case studies.

In Chapter 2, I outline the theoretical framework required to derive hybrid quantum-classical embedding models. After introducing general notation and the second quantization (SQ) formalism (Section 2.1), I briefly describe Hartree-Fock (HF) theory (Section 2.2) and its self-consistent field (SCF) formulation. Then, I show a concise introduction to the ADC scheme for the polarization propagator to obtain electronically excited states (Section 2.3), together with a formulation of ADC within a response theory framework for molecular properties (Section 2.4). In these sections, I use the so-called intermediate state representation (ISR) to express ADC quantities via expectation values.[40] I present a theoretical extension of the existing ADC/ISR framework toward linear response properties of excited states, proposing a simple and universal recipe for deriving ADC/ISR response properties. The last part of the chapter is a detailed derivation of the PE model (Section 2.5). As *the* essential embedding framework of this thesis, I derive all relevant working equations, and I present the formulation of PE within an SCF framework. This ground state PE-SCF scheme is the most basic combination of PE with an electronic structure method, but it is mandatory since higher-level methods build upon it. Chapter 3 contains most of the methodological development achieved in this thesis. First, I present the combination of PE with ADC for excited electronic states employing *a posteriori* perturbative corrections for the excitation energies, called pt-PE-ADC (Section 3.1). This method relies on a self-consistent PE-HF ground state without any explicit contributions to the ADC problem itself, i.e., the environment contributions in ADC only arise indirectly through the modified HF reference. To reduce the resulting error, I add two perturbative corrections, taking into account different physical phenomena, to the zeroth-order excitation energies. Afterwards, I derive a more accurate scheme accounting for dynamic response of the polarizable environment through a linear response coupling term (Section 3.2). This LR-PE-ADC approach retains the computational efficiency of *in vacuo* ADC by using a simplified coupling density. Based on the linear response formalism it is then straightforward to include dynamic environment contributions for response properties within the ADC/ISR framework. The presented scheme enables the usage of PE-ADC in combination with *any* response property available for ADC/ISR. Finally, I derive analytic nuclear PE gradients for correlated ground and excited state methods using the Lagrangian formalism. Next, in Chapter 4, I present the implementation of the three open-source libraries that I use to carry out the implementation of all combined PE-ADC methodologies. The first software component is my standalone library for PE, called CPPE, which can be easily interfaced to any quantum chemical host program. The second component is the `adcc` toolkit which enables rapid development of ADC-based methods. The third library, `respondo`, is a plugin for `adcc` implementing various response properties based on the ADC/ISR framework. The synergy of all three components combined with open-source host programs makes it possible to implement the derived methods in a straightforward manner. In Chapter 5, I apply the derived and implemented PE-ADC methodologies to several case studies, including the investigation of a biomolecular system. In Chapter 6, I present an efficient open-source linear-scaling formulation of the PE model based on the fast multipole method (FMM). This approach is independent of the quantum method coupled to PE and removes the purely classical, quadratically scaling performance bottleneck of PE when treating environments with thousands of polarizable atoms. Naturally, I implement the PE-FMM scheme in the previously developed libraries. In Chapter 7, I discuss the technical and algorithmic details of solving response equations in the ADC/ISR framework. This includes two efficient solver algorithms which I integrate into the `respondo` library and test for convergence behavior. I show numerical case studies of the aforementioned excited state linear response functions in Section 7.3. In Chapter 8, I consider more extreme environments, i.e., the distortion of molecules by the influence of external forces. First, I introduce a new electronic structure method to apply hydrostatic pressure via Gaussian potentials on the single molecule level, called GOSTSHYP (Section 8.1). Second, I show a new computational protocol for quantum mechanochemical analyses of proteins under mechanical stress (Section 8.2). With this protocol I reveal that the force-induced rupture mechanism of the metalloprotein rubredoxin is indeed much more complicated than previously described in the literature. In Chapter 9, I illustrate the design of new photocages built from fluorene derivatives. Together with experimental collaborators, I develop the next-generation fluorene-based photocages with improved absorption behavior and uncaging quantum yields using an efficient computational screening procedure. Finally, I conclude the thesis in Chapter 10 with further ideas and prospects for future research.

Most of the presented results are published in peer-reviewed scientific journals, as indicated by the respective citations and by footnotes at the beginning of the chapters. A complete list of my publications can be found on page 229.

Chapter 2

Theoretical Background

To include effects from complex environments in quantum chemical simulations, the first prerequisite is to build on established electronic structure methods and subsequently adapt and extend these methods accordingly. In this chapter, a brief outline of the most important electronic structure methods for electronic excited states and molecular properties treated in this thesis (Fig. 2.1) is given. The following chapters expand upon these methods and refer to the key equations given herein. In addition, the main environment model used in my work, namely PE, is thoroughly derived for combination with ADC in the next Chapter 3.

2.1 Notation and Second Quantization

Using the second quantization (SQ) formalism, it is possible to define wave functions and operators independent of the particle number. This powerful approach serves to conduct algebraic derivations of matrix elements of operators. For an extensive discussion of SQ, I refer to Refs. 43 and 44, which the description mostly follows.

Consider a basis of spin orbitals $\{\phi_p(\mathbf{r},\sigma)\}$ with the spatial coordinate $\mathbf{r}$ and the spin coordinate σ. In the following, $\mathbf{x}$ is a collective variable of both spatial and spin coordinates, i.e., $\phi_p(\mathbf{r},\sigma) = \phi_p(\mathbf{x})$. Trial wave functions for many-electron systems are typically constructed from *Slater determinants*, which are anti-symmetrized and normalized products

---

Parts of this chapter have already been published in:

- M. Scheurer, T. Fransson, P. Norman, A. Dreuw, and D. R. Rehn, "Complex Excited State Polarizabilities in the ADC/ISR Framework", *J. Chem. Phys.* **2020**, *153*, 074112. (Reference 41)
- M. Scheurer, "Polarizable Embedding for the Algebraic-Diagrammatic Construction Scheme", *Springer Fachmedien Wiesbaden*, **2020**. (Reference 42)

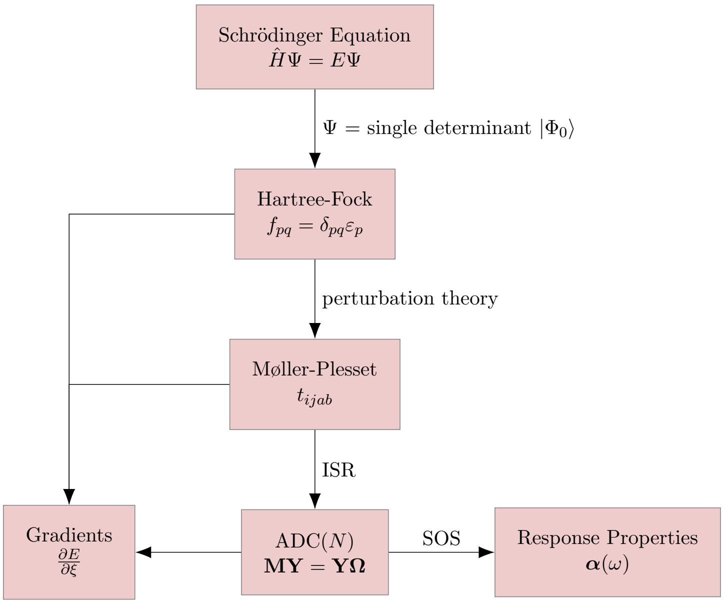


Figure 2.1: Overview of important electronic structure methods in the context of this thesis. The variables and equations are introduced later in the chapter.

of spin orbitals, here for an N-electron system

$$|\Phi\rangle \equiv |\phi_{p_1}\phi_{p_2}\phi_{p_3}\dots\phi_{p_N}| = \frac{1}{\sqrt{N!}}\begin{vmatrix} \phi_{p_1}(\mathbf{x}_1) & \phi_{p_2}(\mathbf{x}_1) & \dots & \dots & \phi_{p_N}(\mathbf{x}_1) \\ \phi_{p_1}(\mathbf{x}_2) & \phi_{p_2}(\mathbf{x}_2) & \dots & \dots & \phi_{p_N}(\mathbf{x}_2) \\ \phi_{p_1}(\mathbf{x}_3) & \phi_{p_2}(\mathbf{x}_3) & \dots & \dots & \phi_{p_N}(\mathbf{x}_3) \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \phi_{p_1}(\mathbf{x}_N) & \phi_{p_2}(\mathbf{x}_N) & \dots & \dots & \phi_{p_N}(\mathbf{x}_N) \end{vmatrix} . \tag{2.1}$$

This ansatz is chosen because it fulfills the Pauli exclusion principle. Such determinants can be defined by means of an abstract linear vector space, called Fock space. Let

$$|\mathbf{k}\rangle = |k_1, k_2, \ \dots, k_m\rangle \, , k_p = \begin{cases} 1 & \phi_p \text{ occupied} \\ 0 & \phi_p \text{ unoccupied} \end{cases} \tag{2.2}$$

be an occupation number (ON) vector that represents a single Slater determinant. Each occupation number k_p indicates whether the spin orbital $\phi_p(\mathbf{x})$ is occupied ($k_p = 1$) or not ($k_p = 0$), i.e., spin orbitals with $k_p = 0$ are not included in the respective Slater determinant. The scalar product of two ON vectors is defined as

$$\langle \mathbf{k} | \mathbf{m} \rangle = \prod_{p=1}^{N} \delta_{k_p m_p}. \tag{2.3}$$

From this definition, it follows that the overlap between Slater determinants with different particle numbers is zero. Note that for a given spin orbital basis, there exists a one-to-one mapping between an ON vector and a given Slater determinant Next, to manipulate determinants or ON vectors, elementary creation and annihilation operators are defined. The creation operator $\hat{a}_p^\dagger$ creates an electron in an unoccupied spin orbital ϕ_p

$$\hat{a}_p^\dagger \left| k_1, k_2, \ \ldots, 0_p, \ \ldots, k_m \right\rangle = \Gamma_p^{\mathbf{k}} \left| k_1, k_2, \ \ldots, 1_p, \ \ldots, k_m \right\rangle. \tag{2.4}$$

The phase factor $\Gamma_p^{\mathbf{k}}$ depends on where in the ON vector the corresponding spin orbital resides. It is -1 if there is an odd number of spin orbitals on the left-hand side of k_p. If there is an even number of spin orbitals on the left-hand side, it is equal to $+1$. Hence, the phase factor gives the correct sign for column permutations in the corresponding Slater determinant. Acting with $\hat{a}_p^\dagger$ on an ON vector where $k_p = 1$, i.e., the spin orbital is already occupied gives zero,

$$\hat{a}_p^\dagger \left| k_1, k_2, \ \ldots, 1_p, \ \ldots, k_m \right\rangle = 0, \tag{2.5}$$

which is in agreement with the Pauli exclusion principle. The complementary annihilation operator $\hat{a}_p$ destroys an electron in spin orbital ϕ_p according to

$$\hat{a}_p \left| \mathbf{k} \right\rangle = \delta_{k_p 1} \Gamma_p^{\mathbf{k}} \left| k_1, k_2, \ \ldots, 0_p, \ \ldots, k_m \right\rangle. \tag{2.6}$$

If $k_p = 0$ and $\hat{a}_p$ acts on the respective ON vector, the result is again zero, because one cannot annihilate a non-existing electron. For derivation of algebraic expressions using the elementary operators, the following anticommutation relations are important:

$$\{\hat{a}_p^\dagger, \hat{a}_q^\dagger\} = 0 \tag{2.7}$$

$$\{\hat{a}_p, \hat{a}_q\} = 0 \tag{2.8}$$

$$\{\hat{a}_p^\dagger, \hat{a}_q\} = \delta_{pq}. \tag{2.9}$$

From these anticommutation relations, all other algebraic properties of SQ can be derived. Representing operators in SQ is a key feature used throughout this thesis. In general, a one-electron operator in first quantization

$$\hat{v}^c = \sum_{i=1}^{N} v^c(\mathbf{r}_i, \sigma_i) \tag{2.10}$$

is represented in SQ by

$$\hat{v} = \sum_{pq} v_{pq} \hat{a}_p^\dagger \hat{a}_q, \tag{2.11}$$

with the integrals of the first-quantized operator in the spin orbital basis

$$v_{pq} = \int \phi_p^*(\mathbf{x}) v^c(\mathbf{x}) \phi_q(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{2.12}$$

The second-quantized representation of the operator does, in contrast to its first-quantized form, not depend on the number of electrons, i.e., no *explicit* electron coordinates are used within the definition of the operator. The spatial structure of the operator, however, enters via the integral matrix elements. If the operator depends only on spatial coordinates, as most often is the case, one can sum over spin coordinates and define

$$\hat{v} = \sum_{pq} v_{pq} \hat{E}_{pq} \tag{2.13}$$

with

$$\hat{E}_{pq} = \hat{a}_{p\sigma}^\dagger \hat{a}_{q\sigma} + \hat{a}_{p\tau}^\dagger \hat{a}_{q\tau}. \tag{2.14}$$

Analogously, one obtains the SQ form of a two-electron operator

$$\hat{g} = \sum_{pqrs} g_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r \tag{2.15}$$

with the four-index two-electron integrals

$$g_{pqrs} = \iint \phi_p^*(\mathbf{x}_1) \phi_q^*(\mathbf{x}_2) g^c(\mathbf{x}_1, \mathbf{x}_2) \phi_r(\mathbf{x}_1) \phi_s(\mathbf{x}_2) \mathrm{d}\mathbf{x}_1 \mathrm{d}\mathbf{x}_2. \tag{2.16}$$

2.2 Hartree-Fock Theory

Hartree-Fock (HF) theory is the cornerstone for more advanced and accurate methods in electronic structure theory, and is thus, despite its simplicity and shortcomings, indispensable in determining an approximate wave function for the electronic ground state.[43–46] The wave function ansatz in HF consists of a *single* Slater determinant $|\Phi_0\rangle$ that minimizes the total energy

$$E_{\mathrm{HF}} = \langle \Phi_0 | \hat{H} | \Phi_0 \rangle \,, \tag{2.17}$$

with the molecular electronic Hamiltonian[43,44]

$$\hat{H} = \hat{h} + \hat{W} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{pqrs} \langle pq | rs \rangle \, \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r . \tag{2.18}$$

The one-electron contribution $\hat{h}$, called *core Hamiltonian*, requires the integrals of the kinetic energy and nuclear attraction operator in the basis of spin orbitals as

$$h_{pq} = -\frac{1}{2} \int \phi_p^*(\mathbf{x}) \nabla^2 \phi_q(\mathbf{x}) \mathrm{d}\mathbf{x} - \sum_K \int \phi_p^*(\mathbf{x}) \frac{Z_K}{|\mathbf{r} - \mathbf{R}_K|} \phi_q(\mathbf{x}) \mathrm{d}\mathbf{r}, \tag{2.19}$$

where the summation runs over all nuclei K with nuclear charge Z_K at position $\mathbf{R}_K$. The two-electron contribution $\hat{W}$ entails the following two-electron integrals of Coulomb repulsion as

$$\langle pq | rs \rangle = \iint \phi_p^*(\mathbf{r}_1) \phi_q^*(\mathbf{r}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \phi_r(\mathbf{r}_1) \phi_s(\mathbf{r}_2) \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 . \tag{2.20}$$

Often, anti-symmetrized two-electron integrals are used[43]

$$\langle pq || rs \rangle = \langle pq | rs \rangle - \langle pq | sr \rangle \,. \tag{2.21}$$

Using the relation

$$\langle pq || rs \rangle = - \langle pq || sr \rangle \tag{2.22}$$

together with

$$\hat{a}_s \hat{a}_r = - \hat{a}_r \hat{a}_s \tag{2.23}$$

one can write the second-quantized Hamiltonian operator in the form

$$\hat{H} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{4} \sum_{pqrs} \langle pq || rs \rangle \, \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r . \tag{2.24}$$

The nuclear repulsion V_{nn} is, within the Born-Oppenheimer approximation,[43,44,47] a constant term given as

$$V_{\mathrm{nn}} = \sum_{K<M} \frac{Z_K Z_M}{|\mathbf{R}_K - \mathbf{R}_M|}, \tag{2.25}$$

which can be added to $\hat{H}$ to yield the total energy of the molecular system. Applying the variational principle for the energy, one can devise *Brioullin's theorem* stating that rotations between occupied orbitals and unoccupied (virtual) orbitals vanish.[44] In other words, singly excited determinant $|\Phi_i^a\rangle$ do not couple to the variationally determined ground state determinant $|\Phi_0\rangle$. Throughout this thesis, let $i, j, k, \ldots$ be the indices of occupied spin orbitals, so-called *hole states*, and $a, b, c, \ldots$ unoccupied spin orbitals, which denote virtual orbitals or *particle states.* As before, $p, q, r, s, \ldots$ stand for general spin orbitals. From the Brioullin condition, an effective one-electron operator, the Fock operator is constructed as

$$\hat{F} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \sum_{pqi} \langle pi||qi\rangle \, \hat{a}_p^\dagger \hat{a}_q \tag{2.26}$$

with matrix elements f_{pq}. The Fock operator matches the exact Hamiltonian $\hat{H}$ in the non-interacting case, i.e., when only the contributions from the core Hamiltonian are included. The second term describes interactions of a single electron in the mean field of all other electrons and depends on the occupied orbitals $\{\phi_i\}$, i.e., the solution of the problem itself. Due to this dependence, one solves the HF problem in an iterative manner until self-consistency is reached. Therefore, such procedures are called self-consistent field (SCF) methods. The spin orbital basis in which the Fock operator is diagonal,

$$f_{pq} = \delta_{pq} \varepsilon_p, \tag{2.27}$$

is called molecular orbital (MO) basis,[44] i.e., the MOs diagonalize the Fock operator. The MO energies ε_p are found as diagonal elements of the Fock operator and are given as

$$\varepsilon_p = h_{pp} + \sum_i \langle ip||ip\rangle \, . \tag{2.28}$$

The HF energy is not equal to the expectation value of the Fock operator, which is the sum of MO energies

$$\langle \Phi_0 | \hat{F} | \Phi_0 \rangle = \sum_i h_{ii} + \sum_{ij} \langle ij || ij \rangle = \sum_i \varepsilon_i, \tag{2.29}$$

but given by the expectation value of the Hamiltonian in eq (2.17) as

$$E_{\mathrm{HF}} = \sum_i h_{ii} + \frac{1}{2} \sum_{ij} \langle ij || ij \rangle = \sum_i \varepsilon_i - \frac{1}{2} \sum_{ij} \langle ij || ij \rangle \,. \tag{2.30}$$

The integral $\langle ij|ij \rangle$ is called *Coulomb* integral since it models the Coulomb repulsion between electrons in different spin orbitals, whereas the *exchange* integral $\langle ij|ji \rangle$ has no analogous interpretation in classical physics.[44,47] It arises due to the fact that electrons are indistinguishable particles. Again, the nuclear repulsion term V_{nn} is here omitted since it does not depend on electron coordinates. Using the above nomenclature, a common partitioning scheme of the Fock operator is

$$\hat{F} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \underbrace{\sum_{pqi} \langle pi|qi \rangle \, \hat{a}_p^\dagger \hat{a}_q}_{\hat{J}} - \underbrace{\sum_{pqi} \langle pi|iq \rangle \, \hat{a}_p^\dagger \hat{a}_q}_{\hat{K}} \tag{2.31}$$

with the Coulomb and exchange operator $\hat{J}$ and $\hat{K}$, respectively.

Basis Set Approximation From a practical point of view, a convenient representation of the MOs needs to be chosen that enables the actual implementation of HF. Usually, MOs are constructed as linear combinations of atomic orbitals (AO), known as LCAO approach. The linear combination of AOs $\{\chi_\mu(\mathbf{r})\}$ gives

$$\phi_p(\mathbf{x}) = \sum_\mu C_{\mu p} \chi_\mu(\mathbf{r}). \tag{2.32}$$

The expansion coefficients $\mathbf{C}$ are called MO coefficients, which also contain the spin coordinate. Unless specified otherwise, Greek letters $\alpha, \beta, \gamma, \ldots, \mu, \nu, \ldots$ are atomic orbital indices. Using the LCAO form of the MOs, one recasts the above equation (2.27) to the AO basis and finds a generalized eigenvalue problem

$$\mathbf{FC} = \mathbf{SC}\boldsymbol{\varepsilon} \;, \quad \mathbf{C}^\dagger \mathbf{SC} = \mathbf{1} \tag{2.33}$$

with the overlap matrix of atomic orbitals

$$S_{\mu\nu} = \int \chi_\mu(\mathbf{r})\chi_\nu(\mathbf{r})\mathrm{d}\mathbf{r}, \tag{2.34}$$

and the orthonormality condition in the MO basis

$$S_{pq} = \delta_{pq}. \tag{2.35}$$

The Fock matrix elements in the AO basis are

$$f_{\mu\nu} = h_{\mu\nu} + \sum_{\gamma\delta} D_{\gamma\delta} \left(\langle \mu\gamma|\nu\delta \rangle - \langle \mu\gamma|\delta\nu \rangle \right) \tag{2.36}$$

introducing the one-particle density matrix $\mathbf{D}$ as

$$D_{\mu\nu} = \sum_i C_{\mu i} C_{\nu i}. \tag{2.37}$$

Transformations between the AO and MO basis are achieved by

$$h_{pq} = \sum_{\mu\nu} C_{\mu p} h_{\mu\nu} C_{\nu q} \tag{2.38}$$

and

$$\langle pq|rs \rangle = \sum_{\mu\nu\gamma\delta} C_{\mu p} C_{\nu q} \langle \mu\nu|\gamma\delta \rangle C_{\gamma r} C_{\delta s}. \tag{2.39}$$

Now, the HF energy can be evaluated by means of the one-particle density matrix as[47]

$$E_{\mathrm{HF}}[\mathbf{D}] = \sum_{\mu\nu} D_{\mu\nu} h_{\mu\nu} + \frac{1}{2} \sum_{\mu\nu\gamma\delta} D_{\mu\nu} D_{\gamma\delta} \left(\langle \mu\gamma|\nu\delta \rangle - \langle \mu\gamma|\delta\nu \rangle \right). \tag{2.40}$$

A graphical illustration of a common SCF algorithm is shown in Figure 2.2. After diagonalization of the Fock operator in each iteration, the density matrix is formed using the aufbau principle,[47] i.e., the energetically lowest molecular orbitals are in the occupied orbital space. Of note, the coefficient matrix, density matrix, and Fock matrix exhibit a

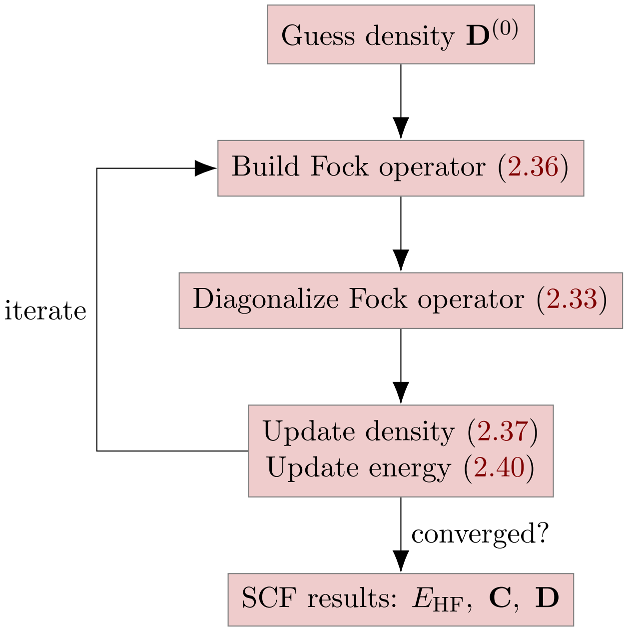


Figure 2.2: Simplified illustration of a typical SCF algorithm. The guess density $\mathbf{D}^{(0)}$ can be obtained by various different methods,[47] for example from the core Hamiltonian.

block structure due to spin

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}^\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^\beta \end{pmatrix} \tag{2.41}$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}^\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^\beta \end{pmatrix} \tag{2.42}$$

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}^\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^\beta \end{pmatrix} \tag{2.43}$$

with the α and β spin parts of each matrix.[47] Off-diagonal blocks are spin-forbidden. In the case where all spatial molecular orbitals are doubly occupied (restricted HF, RHF),[47] the α and β parts are identical and one may reformulate the problem in terms of just one sub-block of the coefficients and density. In the unrestricted case (unrestricted HF, UHF), two individual coefficient matrices, density matrices, and Fock operators are built for the α and β spin part.

Additional Remarks Applying the MO transformations to $\mathbf{D}$, we have that

$$D_{pq} = \sum_{k\mu\nu} C_{\mu p} C_{\mu k} C_{\nu k} C_{\nu q} = \sum_{k} \delta_{kp} \delta_{kq} = \delta_{p\in\text{occ.}} \delta_{q\in\text{occ.}}, \tag{2.44}$$

where $\delta_{p\in\text{occ.}} = 1$ if p is in an occupied orbital space and $\delta_{p\in\text{occ.}} = 0$ otherwise. As follows, the only non-zero block is

$$D_{ij} = \delta_{ij}. \tag{2.45}$$

Obviously, one recovers the HF energy functional (2.30) when writing eq (2.40) in terms of the MO density matrix. Another interesting feature is that the Fock matrix elements can be obtained by differentiating the energy functional with respect to the density matrix, i.e.,

$$f_{\alpha\beta} = \frac{\partial E_{\text{HF}}}{\partial D_{\alpha\beta}} = \sum_{\mu\nu} h_{\mu\nu} \underbrace{\frac{\partial}{\partial D_{\alpha\beta}} D_{\mu\nu}}_{\delta_{\alpha\mu}\delta_{\beta\nu}} + \sum_{\mu\nu\gamma\delta} \underbrace{\frac{\partial}{\partial D_{\alpha\beta}} D_{\mu\nu}}_{\delta_{\alpha\mu}\delta_{\beta\nu}} D_{\gamma\delta} \left(\langle \mu\gamma | \nu\delta \rangle - \langle \mu\gamma | \delta\nu \rangle \right) \tag{2.46}$$

$$= h_{\alpha\beta} + \sum_{\gamma\delta} D_{\gamma\delta} \left(\langle \alpha\gamma | \beta\delta \rangle - \langle \alpha\gamma | \delta\beta \rangle \right). \tag{2.47}$$

2.3 Algebraic-Diagrammatic Construction Scheme

The algebraic-diagrammatic construction scheme for the polarization propagator was originally derived using many-body Green's function theory.[31] In simple terms, the polarization propagator describes the time evolution of polarization within a many-electron system, that is, time-dependent fluctuations of the ground state electron density.[35] The underlying description relies on the spectrum of the electronic Hamiltonian, such that the polarization propagator contains information about the electronic excited states of a molecular system. This becomes directly visible from the spectral representation of the polarization propagator[31,34,35]

$$\Pi_{pq,rs}(\omega) = \underbrace{\sum_{n\neq 0} \frac{\langle \Psi_0 | \hat{a}_q^\dagger \hat{a}_p | \Psi_n \rangle \langle \Psi_n | \hat{a}_r^\dagger \hat{a}_s | \Psi_0 \rangle}{\omega - (E_n - E_0)}}_{\Pi^+_{pq,rs}(\omega)} + \Pi^-_{pq,rs}(\omega), \tag{2.48}$$

where the negative part $\Pi^-_{pq,rs}(\omega)$ contains the same physical information as $\Pi^+_{pq,rs}(\omega)$. The sum-over-states (SOS) expression above possesses poles at the vertical excitation energies $\omega = E_n - E_0$, whereas the residues correspond to transition probabilities of the respective excitation. The polarization propagator can be written in a more compact diagonal matrix form as

$$\Pi(\omega) = \mathbf{x}^\dagger \left(\omega\mathbf{1} - \mathbf{\Omega}\right)^{-1} \mathbf{x}, \tag{2.49}$$

with the diagonal matrix of vertical excitation energies $\Omega_{mn} = (E_n - E_0)\delta_{mn} = \omega_n\delta_{mn}$, and the matrix of transition amplitudes $\mathbf{x}$. Now, a non-diagonal or ADC form of the above matrix equation is postulated

$$\Pi(\omega) = \mathbf{f}^\dagger \left(\omega\mathbf{1} - \mathbf{M}\right)^{-1} \mathbf{f}, \tag{2.50}$$

with the so-called ADC matrix $\mathbf{M}$ and the effective transition moments $\mathbf{f}$. Both the ADC matrix and the effective transition moments are then expanded in a perturbation series such that approximate expressions can be obtained algebraically through a certain order in perturbation theory. An expansion through N-th order in perturbation theory consequently leads to the ADC(N) scheme, which contains all terms needed for a consistent description of $\Pi(\omega)$.[35] The key equation is the resulting Hermitian ADC eigenvalue problem

$$\mathbf{MY} = \mathbf{Y\Omega}, \ \ \mathbf{Y}^\dagger\mathbf{Y} = \mathbf{1}, \tag{2.51}$$

which yields the excitation energies ω_n as the eigenvalues of $\mathbf{M}$ and the corresponding eigenvectors $\mathbf{y}_n$. The ADC matrix exhibits a block structure owing to the perturbation expansions, presented in great detail in existing literature,[31,34,35,48] and illustrated for ADC(0) through ADC(3) in Figure 2.3. The blocks arise from different excitation classes,

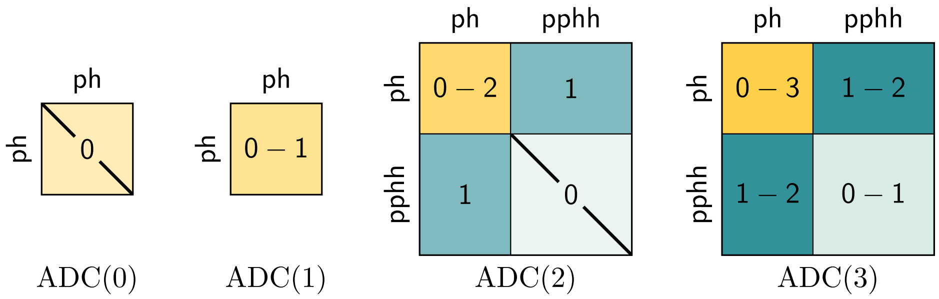


Figure 2.3: ADC matrix $\mathbf{M}$ block structures for ADC(0), ADC(1), ADC(2), and ADC(3). Blocks with only non-zero *diagonal* elements are indicated by a black diagonal line. Illustration adapted from Ref. 49.

i.e., the particle-hole (ph) part describes single excitations, whereas the two-particle-two-hole (pphh) part corresponds to double excitations. Consequently, the ADC excitation vectors possess a ph block and/or pphh block, which correspond to occupied-virtual (ov) and occupied-occupied-virtual-virtual (oovv) orbital spaces, respectively.

An alternative way to derive the ADC equations, which additionally provides access to excited state ADC wave functions and properties, is the so-called intermediate state representation (ISR).[40] Within the ISR, the ADC matrix is a representation of the shifted Hamiltonian

$$M_{IJ} = \langle \tilde{\Psi}_I | \hat{H} - E_0 | \tilde{\Psi}_J \rangle \tag{2.52}$$

in the basis of intermediate states $\{|\tilde{\Psi}_J\rangle\}$. The intermediate state (IS) basis is obtained by applying excitation operators to the correlated ground state wave function and subsequently orthogonalizing the correlated excited states. By applying the ISR formalism to a Møller-Plesset (MP) ground state wave function of order N, one obtains identical perturbation series as with the purely diagrammatic approach as

$$\mathbf{M} = \mathbf{M}^{(0)} + \mathbf{M}^{(1)} + \mathbf{M}^{(2)} + \cdots + \mathbf{M}^{(N)}. \tag{2.53}$$

To obtain the ADC eigenstates, the matrix-vector product of $\mathbf{M}$ with a trial vector is then required to solve eq (2.51) iteratively.[34,35,48] The matrix-vector product equations for the N-th order ADC matrix are thus the key working equations to implement of any ADC method. Using eq (2.51), the IS expansion of the k-th excited state wave function is given by

$$|\Psi_k\rangle = \sum_J Y_{Jk} |\tilde{\Psi}_J\rangle. \tag{2.54}$$

Most importantly, *all* excited state and transition properties can now be evaluated as *expectation values* within the ISR, which makes the combined ADC/ISR scheme attractive from a practical and implementation point of view. In the same spirit as the shifted Hamiltonian, a general one-particle operator $\hat{d}$ (e.g., eq (2.13)) in the IS basis is given by

$$B_{IJ}(\hat{d}) = \langle \tilde{\Psi}_I | \hat{d} | \tilde{\Psi}_J \rangle - \delta_{IJ} \langle \Psi_0 | \hat{d} | \Psi_0 \rangle \,, \tag{2.55}$$

where the ground state expectation value of the operator is subtracted from the diagonal. For transition properties, the so-called modified transition moments are formulated as

$$F_J(\hat{d}) = \langle \tilde{\Psi}_J | \hat{d} \, | \Psi_0 \rangle \,. \tag{2.56}$$

It is also convenient to formulate (transition) properties by means of one-particle density matrices. The transition density matrix from the electronic ground state to an excited state k using the ISR is[34]

$$\rho_{pq}^{k0} = \langle \Psi_k | \hat{a}_p^\dagger \hat{a}_q | \Psi_0 \rangle = \sum_J Y_{Jk}^\dagger \langle \tilde{\Psi}_J | \hat{a}_p^\dagger \hat{a}_q \, | \Psi_0 \rangle \,. \tag{2.57}$$

Density matrices for excited states as well as transition densities between excited states can be computed via

$$\rho_{pq}^{mk} = \langle \Psi_m | \hat{a}_p^\dagger \hat{a}_q | \Psi_k \rangle = \sum_{IJ} Y_{Im}^\dagger \langle \tilde{\Psi}_I | \hat{a}_p^\dagger \hat{a}_q | \tilde{\Psi}_J \rangle Y_{Jk}. \tag{2.58}$$

By setting $m = k$, one obtains the difference density to the ground state

$$\Delta\rho_{pq}^{k} = \rho_{pq}^{kk}. \tag{2.59}$$

A simple contraction with operator integrals yields the desired property, e.g., contracting with the matrix elements of the electric dipole operator gives the transition dipole moments, excited state dipole moments, or transition dipole moments between excited states. Hence, relying on the ADC/ISR formulation directly provides access to a plethora of molecular properties of excited states. A comprehensive summary of the "toolbox" of equations provided by the ADC/ISR framework is shown in Table 2.1.

Table 2.1: ADC/ISR Toolbox Key Equations

Hermitian ADC eigenvalue problem	$\mathbf{MY} = \mathbf{Y\Omega}$, $\mathbf{Y}^\dagger\mathbf{Y} = \mathbf{1}$
Eigenvectors	$\mathbf{Y} = \{\mathbf{y}_n\}$
ADC matrix representation	$M_{IJ} = \langle \tilde{\Psi}_I | \hat{H} - E_0 | \tilde{\Psi}_J \rangle$
IS expansion	$|\Psi_n\rangle = |n\rangle = \sum_J Y_{Jn} |\tilde{\Psi}_J\rangle$
ISR one-particle operator	$B_{IJ}(\hat{d}) = \langle \tilde{\Psi}_I | \hat{d} | \tilde{\Psi}_J \rangle - \delta_{IJ} \langle \Psi_0 | \hat{d} | \Psi_0 \rangle$
Modified transition moments	$F_J(\hat{d}) = \langle \tilde{\Psi}_J | \hat{d} \, | \Psi_0 \rangle$

Going beyond properties that can be directly evaluated as expectation values will be discussed in the following section.

2.4 Response Theory in the ADC/ISR Framework

Response theory offers a general framework to derive and compute a multitude of molecular properties.[2,50] Within ADC, such response properties can elegantly and easily be derived owing to the Hermitian formulation using the formalism of the ISR.[40,51] The ISR offers direct access to excited state wave functions and operators, as explained in the previous section, which makes it straightforward to implement method-independent spectral representations of molecular response functions. Several response properties, e.g., static and frequency-dependent polarizabilities of the electronic ground state,[51–53] two-photon absorption (TPA),[37] and resonant inelastic X-ray scattering (RIXS) cross sections[54] were already derived and implemented using ADC/ISR. In the following paragraphs, I will briefly outline how spectral representations of various response functions can be recast into the corresponding ISR formulation.

2.4.1 ADC Formulation of Linear Response Properties

ADC expressions for response properties were first formulated by Trofimov et al.[51] based on the exact formulation of time-dependent linear response functions by Fetter and Walecka.[55] Consider a property Ω with the associated one-particle operator $\hat{\Omega}$ whose time-dependent linear response shall be expressed in the following. A time-dependent perturbation $\hat{V} f(t)$ with the time-independent perturbation operator $\hat{V}$ is applied to the system, e.g., an oscillating electric field. Note that $f(t) = 0$ for $t < t_0$, i.e., the perturbation is "switched off" before t_0. The system is thus in the unperturbed ground state Ψ_0 for $t \leq t_0$. The time-dependent linear response of Ω is given by[51,55]

$$\Delta\Omega(t) = \int_{-\infty}^{\infty} R_{\Omega,V}(t,t') f(t') \mathrm{d}t'. \tag{2.60}$$

The associated response function is

$$R_{\Omega,V}(t,t') = \theta(t-t') \, \langle\Psi_0|[\hat{V}(t'), \hat{\Omega}(t')]|\Psi_0\rangle \tag{2.61}$$

with the step function $\theta(t-t')$ and the interaction picture form of a one-particle operator[55]

$$\hat{O}(t) = e^{i\hat{H}t} \hat{O} e^{-i\hat{H}t}. \tag{2.62}$$

Parts of this section have already been published in

- <u>M. Scheurer</u>, T. Fransson, P. Norman, A. Dreuw, and D. R. Rehn, "Complex Excited State Polarizabilities in the ADC/ISR Framework", *J. Chem. Phys.* **2020**, *153*, 074112. (Reference 41)

The response function is subsequently transformed to the frequency domain by means of a Fourier transform

$$R_{\Omega,V}(\omega) = \int_{-\infty}^{\infty} e^{i(\omega+i\eta)t} R_{\Omega,V}(t,0)\mathrm{d}t \tag{2.63}$$

yielding the frequency-dependent form

$$R_{\Omega,V}(\omega) = \langle\Psi_0|\hat{\Omega}(\omega - \hat{H} + E_0 + i\eta)^{-1}\hat{V}|\Psi_0\rangle - \langle\Psi_0|\hat{V}(\omega + \hat{H} - E_0 + i\eta)^{-1}\hat{\Omega}|\Psi_0\rangle\,. \tag{2.64}$$

Inserting a resolution of identity for exact states $\mathbf{1} = \sum_n |\Psi_n\rangle\langle\Psi_n|$ in the above expression yields the well-known sum-over-states or spectral representation of the response function

$$R_{\Omega,V}(\omega) = \sum_{n\neq 0} \frac{\langle\Psi_0|\hat{\Omega}|\Psi_n\rangle\langle\Psi_n|\hat{V}|\Psi_0\rangle}{\omega - E_n + E_0 + i\eta} - \frac{\langle\Psi_0|\hat{V}|\Psi_n\rangle\langle\Psi_n|\hat{\Omega}|\Psi_0\rangle}{\omega + E_n - E_0 + i\eta}. \tag{2.65}$$

Clearly, a similarity to the spectral representation of the polarization propagator (2.48) can be seen. From eq (2.64) one can directly obtain the ADC/ISR form of the linear response function by inserting a resolution of identity through intermediate states $\mathbf{1} = \sum_J |\tilde{\Psi}_J\rangle\langle\tilde{\Psi}_J|$ as

$$R_{\Omega,V}(\omega) = \mathbf{F}^\dagger(\hat{\Omega})(\omega - \mathbf{M})^{-1}\mathbf{F}(\hat{V}) - \mathbf{F}^\dagger(\hat{V})(\omega + \mathbf{M})^{-1}\mathbf{F}(\hat{\Omega}). \tag{2.66}$$

Now, consider the response of the A-th Cartesian component of the electric dipole moment $\Omega = \mu_A$ of a system when an external oscillating electric field of the form $-\hat{\mu}_B\mathcal{F}_B(t)$, i.e., $\hat{V} = -\hat{\mu}_B$, is applied. Inserting the operators in (2.66) yields an expression for the linear electric dipole polarizability tensor components[2,51]

$$\alpha_{AB}(\omega) = \mathbf{F}^\dagger(\hat{\mu}_A)(\mathbf{M} - \omega)^{-1}\mathbf{F}(\hat{\mu}_B) + \mathbf{F}^\dagger(\hat{\mu}_B)(\mathbf{M} + \omega)^{-1}\mathbf{F}(\hat{\mu}_A). \tag{2.67}$$

In the static limit, i.e., when applying a static external electric field, the polarizability is

$$\alpha_{AB}(0) = 2\mathbf{F}^\dagger(\hat{\mu}_A)\mathbf{M}^{-1}\mathbf{F}(\hat{\mu}_B). \tag{2.68}$$

2.4.2 From SOS Expressions to the ISR Form

Often, closed-form SOS expressions of response functions are given in textbooks[2] rather than the form using the inverse of the Hamiltonian (2.64). For this reason, I outline how one can convert the textbook SOS expressions to the ADC/ISR form in the following. This is of course nothing new compared to insertion of the resolution of identity above, but just a different way of recasting the SOS form to the ISR form. Let

$$\alpha^0_{AB}(\omega) = \sum_{n\neq 0} \left[\frac{\langle 0|\hat{\mu}_A|n\rangle \langle n|\hat{\mu}_B|0\rangle}{\omega_n - \omega - i\gamma} + \frac{\langle 0|\hat{\mu}_B|n\rangle \langle n|\hat{\mu}_A|0\rangle}{\omega_n + \omega + i\gamma} \right] \tag{2.69}$$

be the SOS expression for the complex electric dipole polarizability of the electronic ground state and the frequency of the incoming oscillating electric field ω.[2] The damping term γ ensures that we have a resonant-convergent SOS expression when $\omega_n = \omega$, and further entails a finite excited state life time. This formulation of damped response theory is also called complex polarization propagator (CPP) approach.[2,54,56] Further, the short-hand notation for wave functions $|\Psi_n\rangle \equiv |n\rangle$ is used. To evaluate the SOS expression as stated above, the entire spectrum of the Hamiltonian is required in order to compute transition dipole moments between the ground state and *all* excited states. The ADC/ISR scheme, however, gives us an approximate representation of the excited state wave functions found in all SOS expressions, such that one can simply insert the ISR expressions for transition moments into the SOS expressions. First, one requires

$$\langle n|\hat{\mu}_A|0\rangle = \sum_I Y^\dagger_{In} \langle \tilde{\Psi}_I|\hat{\mu}_A|0\rangle = \sum_I Y^\dagger_{In} F_I(\hat{\mu}_A) = \mathbf{y}^\dagger_n \mathbf{F}(\hat{\mu}_A) \tag{2.70}$$

which is then inserted into eq (2.69) as

$$\alpha^0_{AB}(\omega) = \sum_{n\neq 0} \left[\frac{\mathbf{F}^\dagger(\hat{\mu}_A)\mathbf{y}_n\mathbf{y}^\dagger_n\mathbf{F}(\hat{\mu}_B)}{\omega_n - \omega - i\gamma} + \frac{\mathbf{F}^\dagger(\hat{\mu}_B)\mathbf{y}_n\mathbf{y}^\dagger_n\mathbf{F}(\hat{\mu}_A)}{\omega_n + \omega + i\gamma} \right]. \tag{2.71}$$

Second, the summation needs to be removed

$$\sum_{n\neq 0} \frac{\mathbf{y}_n\mathbf{y}^\dagger_n}{\omega_n} = \mathbf{Y}\mathbf{\Omega}^{-1}\mathbf{Y}^\dagger. \tag{2.72}$$

This expression can be transformed to the inverse of the ADC matrix using the previous definitions (see Tab. 2.1) according to

$$\begin{aligned}
& \mathbf{M}\mathbf{Y} = \mathbf{Y}\mathbf{\Omega} && \mid \text{multiply with } \mathbf{M}^{-1} \text{ from the left} && (2.73)\\
\Rightarrow\ & \mathbf{Y} = \mathbf{M}^{-1}\mathbf{Y}\mathbf{\Omega} && \mid \text{multiply with } \mathbf{\Omega}^{-1} \text{ from the right} && (2.74)\\
\Rightarrow\ & \mathbf{Y}\mathbf{\Omega}^{-1} = \mathbf{M}^{-1}\mathbf{Y} && \mid \text{multiply with } \mathbf{Y}^\dagger \text{ from the left} && (2.75)\\
\Rightarrow\ & \mathbf{\Omega}^{-1} = \mathbf{Y}^\dagger\mathbf{M}^{-1}\mathbf{Y} && && (2.76)\\
\Rightarrow\ & \mathbf{Y}\mathbf{\Omega}^{-1}\mathbf{Y}^\dagger = \mathbf{Y}\mathbf{Y}^\dagger\mathbf{M}^{-1}\mathbf{Y}\mathbf{Y}^\dagger = \mathbf{M}^{-1}. && && (2.77)
\end{aligned}$$

The other terms in the denominators do not explicitly depend on excitation energies and are just a shift on the diagonal of the ADC matrix. Third, the inverse of the ADC matrix is inserted, where the perturbation frequency is subtracted/added on the diagonal of $\mathbf{M}$, that is

$$\alpha^0_{AB}(\omega) = \mathbf{F}^\dagger(\hat{\mu}_A)(\mathbf{M} - \omega - i\gamma)^{-1}\mathbf{F}(\hat{\mu}_B) + \mathbf{F}^\dagger(\hat{\mu}_B)(\mathbf{M} + \omega + i\gamma)^{-1}\mathbf{F}(\hat{\mu}_A). \tag{2.78}$$

Using the same procedure, all SOS expressions, e.g., for non-linear response properties, are transformable to the ADC/ISR form. An example for a non-linear response property is the TPA transition matrix[37] for an excited state f, which has the following SOS expression

$$S^f_{AB}(\omega) = \sum_k \left[\frac{\langle 0|\hat{\mu}_A|k\rangle \, \langle k|\hat{\mu}_B|f\rangle}{\omega_k - \omega} + \frac{\langle 0|\hat{\mu}_B|k\rangle \, \langle k|\hat{\mu}_A|f\rangle}{\omega_k - \omega} \right]. \tag{2.79}$$

For degenerate TPA, one sets $\omega = \omega_f/2$, such that the energy sum of two degenerate photons yields exactly the excitation energy of the final state f. In the vicinity of two-photon resonances, the TPA matrix yields the so-called TPA transition strength as

$$\delta^f_{\mathrm{TP}} = \frac{1}{15} \sum_{AB} \left(S^f_{AA} S^f_{BB} + S^f_{AB} S^f_{AB} + S^f_{AB} S^f_{BA} \right). \tag{2.80}$$

When converting eq (2.79) to the ISR, special care needs to be taken. First, the ground state $k = 0$ is included in the summation leading to a term with the ground state dipole moment $\mu^{00}_A = \langle 0|\hat{\mu}_A|0\rangle$. This contribution is *not* covered by the modified transition moments as they only cover transitions between the ground and excited states. Consequently, an SOS term that is *missed* by straightforward transformation to the ISR is

$$\Delta^{k=0}_{AB} = \frac{\langle 0|\hat{\mu}_A|0\rangle \, \langle 0|\hat{\mu}_B|f\rangle}{-\frac{\omega_f}{2}} + \frac{\langle 0|\hat{\mu}_B|0\rangle \, \langle 0|\hat{\mu}_A|f\rangle}{-\frac{\omega_f}{2}} = -2\frac{\mu^{00}_A \mu^{0f}_B}{\omega_f} - 2\frac{\mu^{00}_B \mu^{0f}_A}{\omega_f}. \tag{2.81}$$

Second, the dipole moment of the final state $\mu^{ff}_A = \langle f|\hat{\mu}_A|f\rangle$ occurs in the summation. This contribution is, however, not covered via the ISR operator matrix, because the ground state contribution of the operator is subtracted on the diagonal (see eq (2.144))

$$\langle f|\hat{\mu}_A|f\rangle = \mathbf{y}^\dagger_f \mathbf{B}(\hat{\mu}_A)\mathbf{y}_f + \mu^{00}_A, \tag{2.82}$$

whereas the transition moment

$$\langle n|\hat{\mu}_A|f\rangle = \mathbf{n}^\dagger_f \mathbf{B}(\hat{\mu}_A)\mathbf{y}_f \tag{2.83}$$

does not contain a ground state contribution. The missing terms are

$$\Delta_{AB}^{k=f} = \frac{\langle 0|\hat{\mu}_A|f\rangle \langle 0|\hat{\mu}_B|0\rangle}{\frac{\omega_f}{2}} + \frac{\langle 0|\hat{\mu}_B|f\rangle \langle 0|\hat{\mu}_A|0\rangle}{\frac{\omega_f}{2}} = 2\frac{\mu_B^{00}\mu_A^{0f}}{\omega_f} + 2\frac{\mu_A^{00}\mu_B^{0f}}{\omega_f}. \tag{2.84}$$

Intriguingly, the above terms add up to zero

$$\Delta_{AB}^{k=0} + \Delta_{AB}^{k=f} = 0, \tag{2.85}$$

such that no terms are actually ‘missed’ after all, yielding the ISR expression of the TPA matrix as[37]

$$S_{AB}^{f} = \mathbf{F}^{\dagger}(\hat{\mu}_A)\left(\mathbf{M} - \frac{\omega_f}{2}\right)^{-1}\mathbf{B}(\hat{\mu}_B)\mathbf{y}_f + \mathbf{F}^{\dagger}(\hat{\mu}_B)\left(\mathbf{M} - \frac{\omega_f}{2}\right)^{-1}\mathbf{B}(\hat{\mu}_A)\mathbf{y}_f. \tag{2.86}$$

The corresponding SOS expression is often given in the following form

$$S_{AB}^{f} = \sum_{k\neq 0}\left[\frac{\langle 0|\hat{\mu}_A|k\rangle \langle k|\overline{\hat{\mu}}_A|f\rangle}{\omega_k - \omega} + \frac{\langle 0|\hat{\mu}_B|k\rangle \langle k|\overline{\hat{\mu}}_B|f\rangle}{\omega_k - \omega}\right], \tag{2.87}$$

with the fluctuation dipole operator $\overline{\hat{\mu}}_A = \hat{\mu}_A - \mu_A^{00}$. A key point when SOS expressions are converted to the ISR form is to check whether terms in the summation are not covered (ground state or excited state moments) and whether these remaining terms add up to zero or not. Another example for this is the so-called Kramers-Heisenberg-Dirac scattering amplitude for RIXS within the rotating wave approximation[54]

$$\mathcal{F}_{AB}^{0f}(\omega) = \sum_{n}\frac{\langle f|\hat{\mu}_A|n\rangle \langle n|\hat{\mu}_B|0\rangle}{\omega_n - \omega - i\gamma}. \tag{2.88}$$

This expression looks rather similar to the TPA equation (2.79), except that it consists of a single term only. The additional terms not covered by ISR transition moments are given as

$$\Delta_{AB}^{n=f} = \frac{\mu_A^{00}\mu_B^{0f}}{\omega_f - \omega - i\gamma} \tag{2.89}$$

and

$$\Delta_{AB}^{n=0} = \frac{\mu_B^{00}\mu_A^{0f}}{-\omega - i\gamma}. \tag{2.90}$$

These two terms do not cancel each other out and have to be added as constant terms to the ADC/ISR expression of the RIXS amplitude

$$\mathcal{F}_{AB}^{0f}(\omega) = \mathbf{y}_f^\dagger \mathbf{B}(\hat{\mu}_A)(\mathbf{M} - \omega - i\gamma)^{-1}\mathbf{F}(\hat{\mu}_B) + \frac{\mu_A^{00}\mu_B^{0f}}{\omega_f - \omega - i\gamma} + \frac{\mu_B^{00}\mu_A^{0f}}{-\omega - i\gamma}. \tag{2.91}$$

The terms that require extra consideration are relatively easy to find for linear response functions, or when only a single summation index is present. For higher-order properties, the conversion to the ISR form becomes more involved.

2.4.3 Linear Response Functions of Excited States

All previous properties described the response of the ground state to an external perturbation. In the following, I extend the formalism for excited state response properties with the example of complex excited state polarizabilities according to my recent publication.[41] The linear response function (2.64) for a general electronic state $|\Psi_N\rangle$ is given by

$$\begin{aligned} R_{\Omega,V}(\omega) = &\langle\Psi_N|\hat{\Omega}\left(\omega - \hat{H} + E_N\right)^{-1}\hat{V}|\Psi_N\rangle \\ &- \langle\Psi_N|\hat{V}\left(\omega + \hat{H} - E_N\right)^{-1}\hat{\Omega}|\Psi_N\rangle\,. \end{aligned} \tag{2.92}$$

Inserting again $\hat{\Omega} = \hat{\mu}_A$ and $\hat{V} = -\hat{\mu}_B$ as for the ground state case, eq (2.92) can be recast to SOS form to yield the polarizability of an excited state f as

$$\alpha_{AB}^{f}(\omega) = \sum_{n\neq f}\left[\frac{\langle f|\hat{\mu}_A|n\rangle\,\langle n|\hat{\mu}_B|f\rangle}{\omega_n - \omega_f - \omega - i\gamma} + \frac{\langle f|\hat{\mu}_B|n\rangle\,\langle n|\hat{\mu}_A|f\rangle}{\omega_n - \omega_f + \omega + i\gamma}\right]. \tag{2.93}$$

The response function from eq (2.92) was further made resonant-convergent using a complex frequency with the damping term γ in the denominator. Note that special care is required with respect to this SOS expression, because the term where $n = f$ is excluded from the summation. To conveniently express this special summation with an *excluded* excited state, I define the modified quantity

$$\mathbf{M}_f = \mathbf{M} - \omega_f - \omega_f\mathbf{y}_f\mathbf{y}_f^\dagger, \tag{2.94}$$

which shifts the diagonal by ω_f and projects the f-th eigenstate out of the matrix $\mathbf{M}$, and the analogously modified ISR operator matrix

$$\mathbf{B}_f(\hat{d}) = \mathbf{B}(\hat{d}) - \omega_f\mathbf{y}_f\mathbf{y}_f^\dagger. \tag{2.95}$$

These quantities replace the original ADC matrix $\mathbf{M}$ and operator matrix $\mathbf{B}$ in the corresponding excited state response function. As usual, we use eq (2.83) to express the transition moments in eq (2.93). The special case where $n = 0$ yields two additional terms obtained via eq (2.70) as

$$\Delta_{AB}^{n=0} = \frac{\mathbf{y}_f^\dagger \mathbf{F}(\hat{\mu}_A)\mathbf{F}^\dagger(\hat{\mu}_B)\mathbf{y}_f}{-\omega_f - \omega - i\gamma} + \frac{\mathbf{y}_f^\dagger \mathbf{F}(\hat{\mu}_B)\mathbf{F}^\dagger(\hat{\mu}_A)\mathbf{y}_f}{-\omega_f + \omega + i\gamma}. \tag{2.96}$$

This term ensures that the polarizability of the excited state also ‘couples’ to the electronic ground state. Using the modified ISR operator matrix $\mathbf{B}_f$ and adding the additional terms to eq (2.93) we have

$$\begin{aligned}\alpha_{AB}^f(\omega) = \sum_{\substack{n\neq f,\\ n\neq 0}} &\left[\frac{\mathbf{y}_f^\dagger \mathbf{B}_f(\hat{\mu}_A)\mathbf{y}_n \mathbf{y}_n^\dagger \mathbf{B}_f(\hat{\mu}_B)\mathbf{y}_f}{\omega_n - \omega_f - \omega - i\gamma} + \frac{\mathbf{y}_f^\dagger \mathbf{B}_f(\hat{\mu}_B)\mathbf{y}_f \mathbf{y}_f^\dagger \mathbf{B}_f(\hat{\mu}_A)\mathbf{y}_f}{\omega_n - \omega_f + \omega + i\gamma}\right] \\ &+ \frac{\mathbf{y}_f^\dagger \mathbf{F}(\hat{\mu}_A)\mathbf{F}^\dagger(\hat{\mu}_B)\mathbf{y}_f}{-\omega_f - \omega - i\gamma} + \frac{\mathbf{y}_f^\dagger \mathbf{F}(\hat{\mu}_B)\mathbf{F}^\dagger(\hat{\mu}_A)\mathbf{y}_f}{-\omega_f + \omega + i\gamma}. \end{aligned} \tag{2.97}$$

The final step to arrive at a programmable expression is to substitute the excitation vectors $\mathbf{y}_n$ and the denominator with an expression involving the inverse ADC matrix $\mathbf{M}_f^{-1}$, i.e.,

$$\begin{aligned}\alpha_{AB}^f(\omega) = \mathbf{y}_f^\dagger \mathbf{B}_f(\hat{\mu}_A)(\mathbf{M}_f - \omega - i\gamma)^{-1}\mathbf{B}_f(\hat{\mu}_B)\mathbf{y}_f + \mathbf{y}_f^\dagger \mathbf{B}_f(\hat{\mu}_B)(\mathbf{M}_f + \omega + i\gamma)^{-1}\mathbf{B}_f(\hat{\mu}_A)\mathbf{y}_f \\ + \frac{\mathbf{y}_f^\dagger \mathbf{F}(\hat{\mu}_A)\mathbf{F}^\dagger(\hat{\mu}_B)\mathbf{y}_f}{-\omega_f - \omega - i\gamma} + \frac{\mathbf{y}_f^\dagger \mathbf{F}(\hat{\mu}_B)\mathbf{F}^\dagger(\hat{\mu}_A)\mathbf{y}_f}{-\omega_f + \omega + i\gamma}. \end{aligned} \tag{2.98}$$

The expression is almost identical to the ISR form of the ground state complex polarizability (2.78), except that a modified ADC secular matrix and operator matrix are used and two additional terms for coupling to the ground state are needed. The “recipes” to arrive at ADC/ISR expressions from SOS expressions for linear ground and excited state response functions are summarized in the box below.

SOS to ADC/ISR Conversion Recipe

1. Ground state transition moments: $\langle n|\hat{\mu}_A|0\rangle \Rightarrow \mathbf{y}_n^\dagger \mathbf{F}(\hat{\mu}_A)$
2. Excited state transition moments: $\langle n|\hat{\mu}_A|f\rangle \Rightarrow \mathbf{y}_n^\dagger \mathbf{B}(\hat{\mu}_A)\mathbf{y}_f$
3. Resolve spectrum in terms of ADC matrix: $\sum\limits_n \frac{\mathbf{y}_n \mathbf{y}_n^\dagger}{\omega_n - \omega} \Rightarrow (\mathbf{M} - \omega)^{-1}$
4. If $\sum\limits_{n\neq f}$, use modified quantities $\mathbf{M}_f$ and $\mathbf{B}_f$
5. Check for extra terms $\Delta_{AB}^{n=0}$ and $\Delta_{AB}^{n=f}$
6. Add extra terms to final expression if $\Delta_{AB}^{n=0} + \Delta_{AB}^{n=f} \neq 0$

2.4.4 Solving ADC/ISR Response Expressions

To avoid full matrix inversions in the response property expressions presented above, a system of linear equations is defined, here for the general complex case

$$(\mathbf{M} - \omega - i\gamma)\mathbf{x} = \boldsymbol{\mathcal{R}}, \tag{2.99}$$

with a general right-hand side $\boldsymbol{\mathcal{R}}$ and the response vector $\mathbf{x}$. By solving this system of equations in an iterative manner, the resulting response vector can be inserted into the original ADC/ISR expression to yield the desired property. The frequency argument is omitted from the response vector in the following for clarity. To give an example, the first equation for the complex, frequency-dependent excited state polarizability (eq (2.98)) with $\boldsymbol{\mathcal{R}} = \mathbf{B}_f(\hat{\mu}_B)\mathbf{y}_f$ and $\mathbf{M} = \mathbf{M}_f$ is given by

$$(\mathbf{M}_f - \omega - i\gamma)\mathbf{x}_f(\hat{\mu}_B) = \mathbf{B}_f(\hat{\mu}_B)\mathbf{y}_f. \tag{2.100}$$

Subsequently, by inserting the response vector, the polarizability can be evaluated as

$$\begin{aligned}\alpha_{AB}^f(\omega) &= \mathbf{y}_f^\dagger \mathbf{B}_f(\hat{\mu}_A)\mathbf{x}_f(\hat{\mu}_B) + \mathbf{y}_f^\dagger \mathbf{B}_f(\hat{\mu}_B)\mathbf{x}_f(\hat{\mu}_A) \\ &\quad + \frac{\mathbf{y}_f^\dagger \mathbf{F}(\hat{\mu}_A)\mathbf{F}^\dagger(\hat{\mu}_B)\mathbf{y}_f}{-\omega_f - \omega - i\gamma} + \frac{\mathbf{y}_f^\dagger \mathbf{F}(\hat{\mu}_B)\mathbf{F}^\dagger(\hat{\mu}_A)\mathbf{y}_f}{-\omega_f + \omega + i\gamma}.\end{aligned} \tag{2.101}$$

Table 2.2: Examples for response equations with right-hand sides $\mathcal{R}$ and frequency arguments using ADC/ISR.

Property	$\mathcal{R}$	Frequency	Final Expression
$\alpha^0_{AB}(\omega)$ (2.78)	$\mathbf{F}(\hat{\mu}_B)$	$-\omega - i\gamma$	$\mathbf{F}^\dagger(\hat{\mu}_A)\mathbf{x}(\hat{\mu}_B)$
	$\mathbf{F}(\hat{\mu}_A)$	$+\omega + i\gamma$	$\mathbf{F}^\dagger(\hat{\mu}_B)\mathbf{x}(\hat{\mu}_A)$
$S^f_{AB}(\omega)$ (2.86)	$\mathbf{F}(\hat{\mu}_A)$	$-\omega_f/2$	$\mathbf{x}^\dagger(\hat{\mu}_A)\mathbf{B}_f(\hat{\mu}_B)\mathbf{y}_f$
$\mathcal{F}^{0f}_{AB}(\omega)$ (2.91)	$\mathbf{F}(\hat{\mu}_B)$	$-\omega - i\gamma$	$\mathbf{y}^\dagger_f\mathbf{B}(\hat{\mu}_A)\mathbf{x}(\hat{\mu}_B)$

Note that in order to obtain the full polarizability tensor, the response equation has to be solved for the three Cartesian components of the right-hand side and two different frequency argument, amounting to six linear systems in total. Equation (2.99) can be seen as the central building block for all ADC/ISR response properties. Depending on the required property, only the matrix, frequency arguments, and/or right-hand side changes, but the overall form remains the same. How the response vectors are used to evaluate the final tensor in the end also varies. Sometimes, it is sufficient to compute a simple scalar product with another vector (e.g., ground state polarizability), or a matrix multiplication with the ISR operator matrix is needed (e.g., excited state polarizability). The corresponding equations for the above response properties are summarized in Table 2.2. Note that the building blocks among all response properties are virtually identical. This will be exploited when discussing implementation and different algorithmic approaches, together with autogenerated evaluation of ADC/ISR properties in Chapter 7. Furthermore, the ADC matrix in eq (2.99) can be easily modified, making the first combination of ADC/ISR response properties with polarizable environment models in Chapter 3 possible.

2.5 Environment Effects through Polarizable Embedding

In this thesis, the PE model is combined with *ab initio* wave function methods for the computation of excited states, molecular response properties, and nuclear gradients. Hence, PE is the cornerstone for most of the development presented herein. In the following, a comprehensive derivation of PE is shown, mostly following the original literature.[14–16,22,42] For details on how to prepare PE calculations in general, I refer to the excellent tutorial review by Olsen and co-workers.[18]

2.5.1 Derivation of the PE Model

Consider a supermolecular system of two individual, non-overlapping fragments, named A and B, with M_A and M_B nuclei, respectively. The number of fragments does not alter the final equations, and two fragments are treated here for simplicity. In the beginning, there is no distinction between the treatment of fragment A or B. Let

$$\hat{H}^{\mathrm{AB}} = \hat{H}^{\mathrm{A}} + \hat{H}^{\mathrm{B}} + \hat{V}^{\mathrm{AB}} \tag{2.102}$$

be the supermolecular Hamiltonian operator of this composite system with fragment Hamiltonians $\hat{H}^{\mathrm{A}}$ and $\hat{H}^{\mathrm{B}}$, and the interaction Hamiltonian $\hat{V}^{\mathrm{AB}}$. Since the isolated fragment Hamiltonians only contain terms for one specific fragment and *no* interaction, they are identical to the vacuum form of the molecular Hamiltonian in eq (2.24). To arrive at a supermolecular wave function, one assumes that the individual fragment wave functions solve the Schrödinger equation

$$\hat{H}^{\mathrm{A}} |\mathrm{A}\rangle = E^{\mathrm{A}} |\mathrm{A}\rangle \quad \text{and} \quad \hat{H}^{\mathrm{B}} |\mathrm{B}\rangle = E^{\mathrm{B}} |\mathrm{B}\rangle \,. \tag{2.103}$$

Normalization of both wave functions

$$\langle \mathrm{A}|\mathrm{A}\rangle = 1 \quad \text{and} \quad \langle \mathrm{B}|\mathrm{B}\rangle = 1 \tag{2.104}$$

is a further requirement. The fragment wave functions can be expressed by a wave operator, i.e., a string of creation operators, acting on the vacuum state

$$|\mathrm{A}\rangle = \hat{\psi}^{\mathrm{A}} |\mathrm{vac}\rangle \quad \text{and} \quad |\mathrm{B}\rangle = \hat{\psi}^{\mathrm{B}} |\mathrm{vac}\rangle \,. \tag{2.105}$$

Now we write the supermolecular wave function as a simple product of fragment wave functions,

$$|\mathrm{AB}\rangle = \hat{\psi}^{\mathrm{A}}\hat{\psi}^{\mathrm{B}}\,|\mathrm{vac}\rangle\,. \tag{2.106}$$

If all states of each fragment are taken into account, this corresponds to a direct product wave function ansatz, i.e., $|\mathrm{AB}\rangle \in \{\{|A\rangle\} \otimes \{|B\rangle\}\}$. Since the wave operators act on different, non-overlapping molecular fragments, their commutator vanishes

$$\left[\hat{\psi}^{\mathrm{A}}, \hat{\psi}^{\mathrm{B}}\right] = 0. \tag{2.107}$$

Hence, we have

$$[\hat{a}^{\dagger\mathrm{A}}_{p\sigma}, \hat{a}^{\dagger\mathrm{B}}_{q\tau}] = [\hat{a}^{\mathrm{A}}_{p\sigma}, \hat{a}^{\mathrm{B}}_{q\tau}] = [\hat{a}^{\dagger\mathrm{A}}_{p\sigma}, \hat{a}^{\mathrm{B}}_{q\tau}] = 0. \tag{2.108}$$

The last commutator evaluates to zero as it contains vanishing excitation operators as they excite electrons from one fragment to another ($\hat{a}^{\dagger\mathrm{A}}_{p\sigma}\hat{a}^{\mathrm{B}}_{q\tau} = 0$). It is clear that the Hartree product ansatz for the composite wave function only holds if no exchange/overlap between A and B is found. In practical calculations, this assumption is often violated, e.g., in the case where charge transfer between non-overlapping fragments occurs. For these cases, special care is required in order to arrive at physically meaningful results.

The remaining key component is the interaction operator given by

$$\begin{aligned}\hat{V}^{\mathrm{AB}} =& \sum_{m=1}^{M_{\mathrm{B}}} Z^{\mathrm{B}}_m \sum_{pq\in\mathrm{A}} v_{pq}(\mathbf{R}_m)\hat{E}^{\mathrm{A}}_{pq} + \sum_{n=1}^{M_{\mathrm{A}}} Z^{\mathrm{A}}_n \sum_{rs\in\mathrm{B}} v_{rs}(\mathbf{R}_n)\hat{E}^{\mathrm{B}}_{rs} \\ &+ \sum_{\substack{pq\in\mathrm{A}\\ rs\in\mathrm{B}}} v^{\mathrm{AB}}_{pq,rs}\hat{E}^{\mathrm{A}}_{pq}\hat{E}^{\mathrm{B}}_{rs} + \sum_{n=1}^{M_{\mathrm{A}}}\sum_{m=1}^{M_{\mathrm{B}}} \frac{Z^{\mathrm{A}}_n Z^{\mathrm{B}}_m}{|\mathbf{R}_n - \mathbf{R}_m|},\end{aligned} \tag{2.109}$$

with the nuclei-electron attraction between fragments (term 1 and 2), electron-electron repulsion (term 3), and nuclear repulsion (term 4). The nuclear charges of fragment A and B are respectively denoted Z^{A}_n and Z^{B}_m, located at $\mathbf{R}_n$ and $\mathbf{R}_m$. The two-electron excitation operator reduces to a product of two one-electron excitation operators acting on *one* fragment each, as a result of eq (2.107). The general orbital indices p, q here refer to fragment A, whereas r, s are indices of orbitals that belong to fragment B. The integral

$v_{pq}(\mathbf{R})$ is defined as

$$v_{pq}(\mathbf{R}) = -\int \frac{\phi_p^*(\mathbf{r})\phi_q(\mathbf{r})}{|\mathbf{R}-\mathbf{r}|}\mathrm{d}\mathbf{r}. \tag{2.110}$$

These matrix elements are, from a technical point of view, identical to the nuclei-electron attraction operator found in the core Hamiltonian, but with nuclear charges from another molecular system.

The integral over the Coulomb repulsion between electrons in either fragment is given by

$$v_{pq,rs}^{\mathrm{AB}} = \iint \frac{\phi_p^{*\mathrm{A}}(\mathbf{r})\phi_r^{*\mathrm{B}}(\mathbf{r}')\phi_q^{\mathrm{A}}(\mathbf{r})\phi_s^{\mathrm{B}}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'. \tag{2.111}$$

To describe fragment A in the linearly responsive environment of B, interaction energies through Rayleigh-Schrödinger perturbation theory on the supermolecular system are derived. The perturbation operator is then identical to the fragment interaction operator $\hat{V}^{\mathrm{AB}}$. This treatment yields a subsystem Schrödinger equation for fragment A, where the interaction operator is correct through second order.[16] The resulting energies through second order in perturbation theory are given by

$$E^{(0)} = \langle \mathrm{AB}|\hat{H}|\mathrm{AB}\rangle = E^{\mathrm{A}} + E^{\mathrm{B}}, \tag{2.112}$$

$$E^{(1)} = \langle \mathrm{AB}|\hat{V}^{\mathrm{AB}}|\mathrm{AB}\rangle, \tag{2.113}$$

$$E^{(2)} = -\sum_{\substack{ij\\ i+j\neq 0}} \frac{\langle \mathrm{AB}|\hat{V}^{\mathrm{AB}}|\mathrm{A}^i\mathrm{B}^j\rangle \langle \mathrm{A}^i\mathrm{B}^j|\hat{V}^{\mathrm{AB}}|\mathrm{AB}\rangle}{\epsilon_{ij}^{\mathrm{AB}} - \epsilon_{00}^{\mathrm{AB}}}. \tag{2.114}$$

The zeroth-order contribution $E^{(0)}$ is the sum of the unperturbed fragment energies, and the only terms that need further consideration are of higher order. To eliminate the wave function dependence of fragment B, one can recast the interaction Hamiltonian to a Taylor expansion of the inter-fragment Coulomb interactions around the coordinate $\mathbf{R}_{\mathrm{o}}$ in fragment B,

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!}\left(\nabla^k \frac{1}{|\mathbf{r}-\mathbf{R}_{\mathrm{o}}|}\right)\left(\mathbf{r}'-\mathbf{R}_{\mathrm{o}}\right)^k \tag{2.115}$$

with

$$T^{(k)}_{\mathrm{AB}}(\mathbf{r}) = \nabla^k \frac{1}{|\mathbf{r}-\mathbf{R}_{\mathrm{o}}|} \ . \tag{2.116}$$

The summation uses a special multi-index notation with the following properties:

- k is a three-index tuple, $k = (k_x, k_y, k_z)$.
- $|k| = k_x + k_y + k_z$
- $k! = k_x!k_y!k_z!$
- The summation runs over $3^{|k|}$ multi-indices, if no symmetry relations are employed.
- If symmetry is employed, only $\frac{(|k|+1)(|k|+2)}{2}$ unique multi-indices are found, the respective components need an appropriate prefactor.[57]

Further, ∇^k is a short-hand notation for the multi-index power of the partial derivative operator

$$\nabla^k = \left(\frac{\partial}{\partial r_x}\right)^{k_x} \left(\frac{\partial}{\partial r_y}\right)^{k_y} \left(\frac{\partial}{\partial r_z}\right)^{k_z} = \frac{\partial^{|k|}}{\partial \mathbf{r}^k} \ . \tag{2.117}$$

$T^{(k)}_{\mathrm{AB}}(\mathbf{r})$ are elements of the *interaction tensor* (also called T-tensor),[57] where (k) specifies the index of multiple tensor elements and the rank of the tensor through $|k|$. More details on the T-tensors can be found in Ref. 57. The Taylor expansion yields the Cartesian multipole moment operator $(\mathbf{r}'-\mathbf{R}_{\mathrm{o}})^k$. The expressions for all Taylor expansions of the Coulomb interaction operator are given by

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \left(\nabla^k \frac{1}{|\mathbf{r}-\mathbf{R}_{\mathrm{o}}|}\right) (\mathbf{r}'-\mathbf{R}_{\mathrm{o}})^k = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} T^{(k)}_{\mathrm{AB}}(\mathbf{r}) (\mathbf{r}'-\mathbf{R}_{\mathrm{o}})^k \tag{2.118}$$

$$\frac{1}{|\mathbf{r}-\mathbf{R}_m|} = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} T^{(k)}_{\mathrm{AB}}(\mathbf{r}) (\mathbf{R}_m-\mathbf{R}_{\mathrm{o}})^k \tag{2.119}$$

$$\frac{1}{|\mathbf{R}_n - \mathbf{r}'|} = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} T^{(k)}_{\mathrm{AB}}(\mathbf{R}_n)\left(\mathbf{r}' - \mathbf{R}_{\mathrm{o}}\right)^k \tag{2.120}$$

$$\frac{1}{|\mathbf{R}_n - \mathbf{R}_m|} = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} T^{(k)}_{\mathrm{AB}}(\mathbf{R}_n)(\mathbf{R}_m - \mathbf{R}_{\mathrm{o}})^k \tag{2.121}$$

Using these expressions, we can write the interaction Hamiltonian as

$$\begin{aligned}
\hat{V}^{\mathrm{AB}} = {} & \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \underbrace{\sum_{m=1}^{M_{\mathrm{B}}} Z^{\mathrm{B}}_m \left(\mathbf{R}_m - \mathbf{R}_{\mathrm{o}}\right)^k}_{\mathcal{Q}^{(k)}_{\mathrm{B,nuc}}} \underbrace{\sum_{pq\in\mathrm{A}} \left(- \int \phi^*_p(\mathbf{r}) T^{(k)}_{\mathrm{AB}}(\mathbf{r}) \phi_q(\mathbf{r}) \mathrm{d}\mathbf{r} \right) \hat{E}^{\mathrm{A}}_{pq}}_{\hat{\mathcal{V}}^{(k)}_{\mathrm{A,el}}} \\
& + \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \underbrace{\sum_{n=1}^{M_{\mathrm{A}}} Z^{\mathrm{A}}_n T^{(k)}_{\mathrm{AB}}(\mathbf{R}_n)}_{\mathcal{V}^{(k)}_{\mathrm{A,nuc}}} \underbrace{\sum_{rs\in\mathrm{B}} \left(- \int \phi^*_r(\mathbf{r}') \left(\mathbf{r}' - \mathbf{R}_{\mathrm{o}}\right)^k \phi_s(\mathbf{r}') \mathrm{d}\mathbf{r}' \right) \hat{E}^{\mathrm{B}}_{rs}}_{\hat{\mathcal{Q}}^{(k)}_{\mathrm{B,el}}} \\
& + \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \underbrace{\sum_{pq\in\mathrm{A}} \left(- \int \phi^*_p(\mathbf{r}) T^{(k)}_{\mathrm{AB}}(\mathbf{r}) \phi_q(\mathbf{r}) \mathrm{d}\mathbf{r} \right) \hat{E}^{\mathrm{A}}_{pq}}_{\hat{\mathcal{V}}^{(k)}_{\mathrm{A,el}}} \times \\
& \qquad \underbrace{\sum_{rs\in\mathrm{B}} \left(- \int \phi^*_r(\mathbf{r}') \left(\mathbf{r}' - \mathbf{R}_{\mathrm{o}}\right)^k \phi_s(\mathbf{r}') \mathrm{d}\mathbf{r}' \right) \hat{E}^{\mathrm{B}}_{rs}}_{\hat{\mathcal{Q}}^{(k)}_{\mathrm{B,el}}} \\
& + \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \underbrace{\sum_{n=1}^{M_{\mathrm{A}}} Z^{\mathrm{A}}_n T^{(k)}_{\mathrm{AB}}(\mathbf{R}_n)}_{\mathcal{V}^{(k)}_{\mathrm{A,nuc}}} \underbrace{\sum_{m=1}^{M_{\mathrm{B}}} Z^{\mathrm{B}}_m \left(\mathbf{R}_m - \mathbf{R}_{\mathrm{o}}\right)^k}_{\mathcal{Q}^{(k)}_{\mathrm{B,nuc}}}.
\end{aligned} \tag{2.122}$$

Collecting terms for each fragment, one finds

$$\hat{V}^{\mathrm{AB}} = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \left(\mathcal{V}^{(k)}_{\mathrm{A,nuc}} + \hat{\mathcal{V}}^{(k)}_{\mathrm{A,el}} \right) \left(\mathcal{Q}^{(k)}_{\mathrm{B,nuc}} + \hat{\mathcal{Q}}^{(k)}_{\mathrm{B,el}} \right) = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \hat{\mathcal{V}}^{(k)}_{\mathrm{A}} \hat{\mathcal{Q}}^{(k)}_{\mathrm{B}}. \tag{2.123}$$

Plugging in the new definition of the interaction Hamiltonian into eq (2.113), the first-order energy correction is

$$E^{(1)} = E_{\mathrm{es}}^{\mathrm{AB}} = \sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \left(\mathcal{V}_{\mathrm{A,nuc}}^{(k)} + \langle \mathrm{A} | \hat{\mathcal{V}}_{\mathrm{A,el}}^{(k)} | \mathrm{A} \rangle \right) \mathcal{Q}_{\mathrm{B}}^{(k)} . \tag{2.124}$$

Here, the expectation value of the multipole moment operator $\mathcal{Q}_{\mathrm{B}}^{(k)}$ is evaluated on fragment B. Thus, the first-order energy describes the plain electrostatic interaction between the fragments in terms of Cartesian multipole moments of fragment B. From this expression, one can directly obtain the effective PE operator for permanent electrostatics.

The second-order energy needs further analysis: The sum-over-states expression requires at least one system or both systems at once to be in an electronically excited state. Hence, the second-order energy can be split up into three individual contributions as

$$\begin{aligned} E^{(2)} &= -\sum_{i \neq 0} \frac{\langle \mathrm{AB} | \hat{V}^{\mathrm{AB}} | \mathrm{A}^{i}\mathrm{B} \rangle \langle \mathrm{A}^{i}\mathrm{B} | \hat{V}^{\mathrm{AB}} | \mathrm{AB} \rangle}{\epsilon_{i}^{\mathrm{A}} - \epsilon_{0}^{\mathrm{A}}} \\ &\quad - \sum_{j \neq 0} \frac{\langle \mathrm{AB} | \hat{V}^{\mathrm{AB}} | \mathrm{AB}^{j} \rangle \langle \mathrm{AB}^{j} | \hat{V}^{\mathrm{AB}} | \mathrm{AB} \rangle}{\epsilon_{j}^{\mathrm{B}} - \epsilon_{0}^{\mathrm{B}}} + E_{\mathrm{disp}} \qquad (2.125) \\ &= E_{\mathrm{ind}}^{\mathrm{A}} + E_{\mathrm{ind}}^{\mathrm{B}} + E_{\mathrm{disp}} \; . \qquad (2.126) \end{aligned}$$

The induction energy of fragment A, $E_{\mathrm{ind}}^{\mathrm{A}}$, does not need to be taken into account since it is implicitly included through the effective PE operator in the wave function optimization such that $|\mathrm{A}\rangle$ is being polarized by the multipole moments of fragment B. Furthermore, the aforementioned induction energy is a classical term, which is contained in the electron density of the molecular system through the orbital rotations that build the electronic ground state. The last term in eq (2.126), E_{disp}, denotes the dispersion energy (both systems are in an excited state).[14,15,58] Since there is no well-defined way to include this energy contribution in operator form, it is neglected. Note that an empirical dispersion correction using a Lennard-Jones potential between fragment A and B can be introduced in an ad hoc manner.[16,59]

Finally, we can evaluate the expectation value of the interaction Hamiltonian of the second term in eq (2.126), namely when fragment B is in an excited state and fragment

A is in the electronic ground state. This yields

$$E_{\mathrm{ind}}^{\mathrm{B}} = -\sum_{|k|=0}^{\infty} \frac{(-1)^{|k|}}{k!} \langle \mathrm{A}|\hat{\mathcal{V}}_{\mathrm{A}}^{(k)}|\mathrm{A}\rangle \sum_{j\neq 0} \frac{\langle \mathrm{B}|\hat{\mathcal{Q}}_{\mathrm{B}}^{(k)}|\mathrm{B}^j\rangle \langle \mathrm{B}^j|\hat{\mathcal{Q}}_{\mathrm{B}}^{(k)}|\mathrm{B}\rangle}{\epsilon_j^{\mathrm{B}} - \epsilon_0^{\mathrm{B}}} \langle \mathrm{A}|\hat{\mathcal{V}}_{\mathrm{A}}^{(k)}|\mathrm{A}\rangle \,. \tag{2.127}$$

The sum-over-states expression describes the polarizabilities of fragment B (eq (2.69)). The zeroth-order term does not contribute due to orthogonality of the states, i.e., $\langle \mathrm{B}|\mathrm{B}^j\rangle = 0$. When truncating the expansion at first order $|k| = 1$, and evaluating the expectation values on fragment A, one finds

$$E_{\mathrm{ind}}^{\mathrm{B}} = -\frac{1}{2}\left(\boldsymbol{\mathcal{F}}_{\mathrm{nuc}}^{\mathrm{A}} + \boldsymbol{\mathcal{F}}_{\mathrm{el}}^{\mathrm{A}}\right) \boldsymbol{\alpha}^{\mathrm{B}} \left(\boldsymbol{\mathcal{F}}_{\mathrm{nuc}}^{\mathrm{A}} + \boldsymbol{\mathcal{F}}_{\mathrm{el}}^{\mathrm{A}}\right) = -\frac{1}{2}\left(\boldsymbol{\mathcal{F}}_{\mathrm{nuc}}^{\mathrm{A}} + \boldsymbol{\mathcal{F}}_{\mathrm{el}}^{\mathrm{A}}\right) \boldsymbol{\mu}_{\mathrm{ind}}^{\mathrm{B}}(\boldsymbol{\mathcal{F}}_{\mathrm{tot}}). \tag{2.128}$$

Here, the total electric field arises from the electrons, $\boldsymbol{\mathcal{F}}_{\mathrm{el}}^{\mathrm{A}}$, and nuclei, $\boldsymbol{\mathcal{F}}_{\mathrm{nuc}}^{\mathrm{A}}$, of the quantum region, however, the expression can be extended for multiple expansion sites in the environment, as commonly done in practice and outlined in the following section.

PE Ingredients

1. Assume non-overlapping fragments
2. Hartree-product ansatz for super-system wave function
3. Rayleigh-Schrödinger perturbation theory (linearly responsive environment)
4. Multipole expansion on one of the fragments
5. Keep interactions representable in operator form (discard dispersion interaction)

2.5.2 SCF Framework for PE

In the following, an effective PE operator is derived to couple PE to the SCF problem. The equations derived above are generalized for S multipole expansion sites in the environment. The resulting multi-center multipole expansion in practice yields better convergence for the short range interactions. The electrostatics part of the operator is easily generalized for multiple sites in the environment, whereas for the induction part, one needs to handle the coupled linear equations for induced dipole moments.

The total PE energy functional is given by[14–16]

$$E_{\text{tot}} = E_{\text{QM}} + E_{\text{PE}} + E_{\text{env}}, \tag{2.129}$$

where E_{QM} is the energy of the quantum region, including polarization of the wave function. E_{PE} represents the interaction energy of the quantum region and the environment, including polarization of the environment. E_{env} denotes the internal energy of all fragments in the environment, including interaction among fragments but excluding energy contributions from polarization in the environment. Decomposing the polarizable embedding energy, one finds

$$E_{\text{PE}} = E_{\text{es}} + E_{\text{ind}}, \tag{2.130}$$

where E_{es} denotes the permanent electrostatic interaction between the core quantum part and the fragments in the environment, and E_{ind} is the energy due to induced charge distributions of the environment. The explicit equation for the electrostatic interaction energy is

$$E_{\text{es}} = E_{\text{es}}^{\text{nuc}} + E_{\text{es}}^{\text{el}}, \tag{2.131}$$

comprised by the nuclear and electronic interaction energies, respectively, where the former is given by

$$E_{\text{es}}^{\text{nuc}} = \sum_{s=1}^{S} \sum_{|k|=0}^{K_s} \frac{(-1)^{|k|}}{k!} Q_s^{(k)} \sum_{n=1}^{N} T_{sn}^{(k)} Z_n. \tag{2.132}$$

The summation over $|k|$ is running over $3^{|k|}$ multi-indices up to the truncation level K_s of the multipole expansion and the summation over s is running over the S sites in the environment. The $Q_s^{(k)}$ is thus the k-th component of the $|k|$-th-order Cartesian multipole moment located at the expansion site coordinate $\mathbf{R}_s$ in the environment, and Z_n is the nuclear charge of the n-th nucleus in the quantum region comprised of N nuclei. As before, we have used the k-th component of the interaction tensor, $T_{ij}^{(k)}$, between two sites i and j,

$$T_{ij}^{(k)} = \frac{\partial^{|k|}}{\partial x_j^{k_x} \partial y_j^{k_y} \partial z_j^{k_z}} \left(\frac{1}{|\mathbf{r}_j - \mathbf{r}_i|} \right). \tag{2.133}$$

Further, the electrostatic interaction energy of the electrons with the environment is given by

$$E_{\mathrm{es}}^{\mathrm{el}} = \langle \Phi_0 | \hat{V}^{\mathrm{es}} | \Phi_0 \rangle = \sum_{pq} D_{pq} V_{pq}^{\mathrm{es}}. \tag{2.134}$$

Using the SQ formalism, one can write the electrostatic operator $\hat{V}^{\mathrm{es}}$ as

$$\hat{V}^{\mathrm{es}} = \sum_{s=1}^{S} \sum_{|k|=0}^{K_s} \frac{(-1)^{|k|}}{k!} Q_s^{(k)} \sum_{pq} t_{pq}^{(k)}(\mathbf{R}_s) \hat{E}_{pq}, \tag{2.135}$$

with the one-electron orbital excitation operator $\hat{E}_{pq}$ and general molecular orbital indices p and q. The integrals are given by

$$t_{pq}^{(k)}(\mathbf{R}_s) = -\int \phi_p^*(\mathbf{r}_1) T_{s1}^{(k)} \phi_q(\mathbf{r}_1) \mathrm{d}\mathbf{r}_1, \tag{2.136}$$

and include again the k-th component of the interaction tensor (eq (2.133)).

The induction energy contribution of a linearly responsive environment amounts to

$$E_{\mathrm{ind}} = -\frac{1}{2} \sum_{s=1}^{S} \boldsymbol{\mu}_{\mathrm{ind},s} \boldsymbol{\mathcal{F}}_s, \tag{2.137}$$

where $\boldsymbol{\mu}_{\mathrm{ind},s}$ is the induced dipole moment at site s in the environment, and $\boldsymbol{\mathcal{F}}_s$ is the electric field vector acting on site s, comprising the field from nuclei and electrons, as well as the fields caused by the permanent multipole moments, i.e.,

$$\boldsymbol{\mathcal{F}}[\mathbf{D}] = \boldsymbol{\mathcal{F}}_{\mathrm{nuc}} + \boldsymbol{\mathcal{F}}_{\mathrm{el}}[\mathbf{D}] + \boldsymbol{\mathcal{F}}_{\mathrm{mul}} \ . \tag{2.138}$$

The static fields produced by the nuclei in the quantum region at site s in the environment are given as

$$\boldsymbol{\mathcal{F}}_{\mathrm{nuc},s} = -\sum_{n=1}^{N} Z_n \mathbf{T}_{ns}^{(1)}, \tag{2.139}$$

whereas the static multipole fields created at site s are[a]

$$\boldsymbol{\mathcal{F}}_{\mathrm{mul},s} = \sum_{s' \neq s} \sum_{|k|=0}^{K_{s'}} \frac{(-1)^{|k|+1}}{k!} Q_{s'}^{(k)} \mathbf{T}_{ss'}^{(k+1)}. \tag{2.140}$$

Note that the electric field from the electrons, and in turn the total field vector, depend on the electronic density matrix $\mathbf{D}$. We find that the induced moment at a site s depends on the total electric field and is given by

$$\boldsymbol{\mu}_{\mathrm{ind},s} = \boldsymbol{\alpha}_s \left(\boldsymbol{\mathcal{F}}_s[\mathbf{D}] + \boldsymbol{\mathcal{F}}_{\mathrm{ind},s} \right). \tag{2.141}$$

The induced fields are produced by all other polarizable sites through

$$\boldsymbol{\mathcal{F}}_{\mathrm{ind},s} = \sum_{s' \neq s} \mathbf{T}_{ss'}^{(2)} \boldsymbol{\mu}_{\mathrm{ind},s'} \ . \tag{2.142}$$

This leads to a linear system of equations,

$$\mathbf{B} \boldsymbol{\mu}_{\mathrm{ind}} = \boldsymbol{\mathcal{F}}[\mathbf{D}], \tag{2.143}$$

with the classical response matrix $\mathbf{B}$,[60] given by

$$\mathbf{B} = \begin{pmatrix} \boldsymbol{\alpha}_1^{-1} & -\mathbf{T}_{12}^{(2)} & \cdots & -\mathbf{T}_{1S}^{(2)} \\ -\mathbf{T}_{21}^{(2)} & \boldsymbol{\alpha}_2^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\mathbf{T}_{(S-1)S}^{(2)} \\ -\mathbf{T}_{S1}^{(2)} & \cdots & -\mathbf{T}_{S(S-1)}^{(2)} & \boldsymbol{\alpha}_S^{-1} \end{pmatrix}. \tag{2.144}$$

The inverse polarizability tensors $\boldsymbol{\alpha}_s^{-1}$ are on the diagonal and the dipole-dipole interaction tensors reside on off-diagonal elements. Subsequently, one can include the induced dipole field into the wave function optimization through the induction operator

$$\hat{V}^{\mathrm{ind}}[\mathbf{D}] = -\sum_{s=1}^{S} \sum_{a=x,y,z} \mu_{\mathrm{ind},s,a}[\mathbf{D}] \hat{\mathcal{F}}_a(\mathbf{R}_s) = -\boldsymbol{\mu}_{\mathrm{ind}}[\mathbf{D}] \hat{\boldsymbol{\mathcal{F}}} = -\hat{\boldsymbol{\mathcal{F}}} \mathbf{B}^{-1} \boldsymbol{\mathcal{F}}[\mathbf{D}], \tag{2.145}$$

using a for the respective Cartesian component x, y, or z. Further, we define the electric-

---

[a] The bold notation of the T-tensor symbolizes the three uncontracted components which are the elements of the resulting electric field vector.

field operator as

$$\hat{\mathcal{F}}_a(\mathbf{R}_s) = \sum_{pq} t_{a,pq}(\mathbf{R}_s)\hat{E}_{pq}. \tag{2.146}$$

The electric-field integrals are

$$t_{a,pq}(\mathbf{R}_s) = -\int \phi_p^*(\mathbf{r})\frac{R_{s,a} - r_a}{|\mathbf{R}_s - \mathbf{r}|^3}\phi_q(\mathbf{r})\mathrm{d}\mathbf{r}. \tag{2.147}$$

and the electric field operator $\hat{\boldsymbol{\mathcal{F}}}$ contains the electric field operators for each site s and component a

$$\hat{\boldsymbol{\mathcal{F}}} = \begin{pmatrix} \hat{\mathcal{F}}_x(\mathbf{R}_1) \\ \hat{\mathcal{F}}_y(\mathbf{R}_1) \\ \vdots \\ \hat{\mathcal{F}}_x(\mathbf{R}_2) \\ \vdots \\ \hat{\mathcal{F}}_z(\mathbf{R}_S) \end{pmatrix}. \tag{2.148}$$

The matrix elements of $\hat{\boldsymbol{\mathcal{F}}}$ are defined as follows

$$\boldsymbol{\mathcal{F}}_{pq} = \begin{pmatrix} t_{x,pq}(\mathbf{R}_1) \\ t_{y,pq}(\mathbf{R}_1) \\ \vdots \\ t_{x,pq}(\mathbf{R}_2) \\ \vdots \\ t_{z,pq}(\mathbf{R}_S) \end{pmatrix}. \tag{2.149}$$

Finally, one solves the SCF problem (2.33) in the presence of the total embedding operator,

$$\hat{V}^{\mathrm{PE}}[\mathbf{D}] = \hat{V}^{\mathrm{es}} + \hat{V}^{\mathrm{ind}}[\mathbf{D}], \tag{2.150}$$

from which we obtain an effective Fock operator

$$\hat{F}^{\mathrm{PE}} = \hat{F} + \hat{V}^{\mathrm{PE}}. \tag{2.151}$$

Since the embedding operator depends on the one-particle density matrix $\mathbf{D}$ itself, namely through the electric fields, the overall embedding operator is non-linear. As for a usual

SCF procedure, the embedding operator is updated in every iteration using the current SCF density matrix. Thus, polarization effects are treated in a fully self-consistent manner for the SCF reference state.

MO representation of the PE operator It is useful to take a close look at the properties of the PE contribution to the Fock operator for derivations following in Chapter 3. Transforming the operator (2.150) to the MO basis, one has the following matrix elements

$$V_{pq}^{\text{PE}} = V_{pq}^{\text{es}} + \underbrace{\sum_i -\boldsymbol{\mathcal{F}}_{pq}\mathbf{B}^{-1}\left(\boldsymbol{\mathcal{F}}_{ii} + \boldsymbol{\mathcal{F}}_{\text{nuc}} + \boldsymbol{\mathcal{F}}_{\text{mul}}\right)}_{V_{pq}^{\text{ind}}}. \tag{2.152}$$

From the equation above, the similarity of the induction operator and Coulomb/exchange operators in eq (2.31) becomes evident: Both require a summation over the occupied orbitals, and one can write the integrals of the induction operator in a more general four-index form

$$V_{pqrs}^{\text{ind}} = -\boldsymbol{\mathcal{F}}_{pq}\mathbf{B}^{-1}\left(\boldsymbol{\mathcal{F}}_{rs} + \boldsymbol{\mathcal{F}}_{\text{nuc}} + \boldsymbol{\mathcal{F}}_{\text{mul}}\right). \tag{2.153}$$

such that the PE operator is more compactly written as

$$V_{pq}^{\text{PE}} = V_{pq}^{\text{es}} + \sum_i V_{pqii}^{\text{ind}}. \tag{2.154}$$

2.5.3 PE-SCF Nuclear Gradients

For geometry optimization or dynamics within the PE framework, analytical nuclear gradients for the quantum region molecule are needed.[59] Since the PE contribution is described via the Fock operator, one may proceed just like in the vacuum case. This means that, in addition to the vacuum contributions, only explicit PE energy terms need to be considered. The derivation is particularly easy because a) the quantum region coordinates are not coupled to the expansion sites in the environment and b) the induced dipole moments fulfill the variational condition. Starting from eq (2.129) and differentiating with respect to nuclear displacement $\mathbf{R}_a$, indicated with the superscript a, we have

$$\mathbf{E}_{\text{tot}}^{a} = \frac{\partial E_{\text{tot}}}{\partial \mathbf{R}_a} = \frac{\partial E_{\text{QM}}}{\partial \mathbf{R}_a} + \frac{\partial E_{\text{PE}}}{\partial \mathbf{R}_a} + \underbrace{\frac{\partial E_{\text{env}}}{\partial \mathbf{R}_a}}_{0}. \tag{2.155}$$

The first term is equal to the vacuum SCF gradient, and the last term vanishes because it only depends on the (fixed) positions of the expansion sites in the environment. Note that $\mathbf{E}^a_{\text{tot}}$ is a vector with three elements, one for each Cartesian coordinate. The second term is split up into three individual parts

$$\mathbf{E}^a_{\text{PE}} = \mathbf{E}^{\text{nuc},a}_{\text{es}} + \mathbf{E}^{\text{el},a}_{\text{es}} + \mathbf{E}^a_{\text{ind}}. \tag{2.156}$$

The nuclear part of the electrostatics contribution (derivative of eq (2.132)) expands to

$$E^{\text{nuc},a_l}_{\text{es}} = \sum_{s=1}^{S} \left(\sum_{|k|=0}^{K_s} \frac{(-1)^{|k|}}{k!} Q^{(k)}_s T^{(k+l)}_{sa} Z_a \right) \tag{2.157}$$

where l is a multi-index denoting the Cartesian component of the derivative with $|l| = 1$. As a result, T-tensors of order $|k|+1$ are needed, but the additional Cartesian components are assigned to the derivative itself and are not summed over. The electronic part of the electrostatics contribution is

$$E^{\text{el},a_l}_{\text{es}} = \sum_{s=1}^{S} \sum_{|k|=0}^{K_s} \frac{(-1)^{|k|}}{k!} Q^{(k)}_s \sum_{pq} t^{(k)a_l}_{pq}(\mathbf{R}_s) D_{pq}, \tag{2.158}$$

with the integral derivatives

$$t^{(k)a_l}_{pq}(\mathbf{R}_s) = - \left(\langle \phi^{a_l}_p | T^{(k)}_{si} | \phi_q \rangle + \langle \phi_p | T^{(k)}_{si} | \phi^{a_l}_q \rangle \right). \tag{2.159}$$

Therein, $\phi^{a_l}_p$ denotes the derivative of the orbital ϕ_p with respect to the nuclear coordinate $\mathbf{R}_a$ and its Cartesian component l. Note that the mid center of the above integral, i.e., the T-tensor itself, is independent of nuclear displacement such that no additional term through product differentiation arises. The last term needed is the derivative of the induction energy

$$\begin{align} \mathbf{E}^a_{\text{ind}} &= -\frac{1}{2} \frac{\partial}{\partial \mathbf{R}_a} \left(\boldsymbol{\mathcal{F}} \mathbf{B}^{-1} \boldsymbol{\mathcal{F}} \right) \tag{2.160} \\ &= -\frac{1}{2} \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{R}_a} \mathbf{B}^{-1} \boldsymbol{\mathcal{F}} - \frac{1}{2} \boldsymbol{\mathcal{F}} \mathbf{B}^{-1} \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{R}_a} \tag{2.161} \\ &= -\boldsymbol{\mu}_{\text{ind}} \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{R}_a} = -\boldsymbol{\mu}_{\text{ind}} \left(\boldsymbol{\mathcal{F}}^a_{\text{nuc}} + \boldsymbol{\mathcal{F}}^a_{\text{el}} \right). \tag{2.162} \end{align}$$

No total derivative of the induced dipole moments with respect to nuclear coordinates is required because the induced moments are variationally determined and minimize the induction energy, i.e.,

$$\frac{\partial E_{\mathrm{ind}}}{\partial \boldsymbol{\mu}_{\mathrm{ind}}} \frac{\mathrm{d}\boldsymbol{\mu}_{\mathrm{ind}}}{\mathrm{d}\mathbf{R}_a} = 0. \tag{2.163}$$

The derivative of electric fields generated by nuclear charges is

$$\boldsymbol{\mathcal{F}}^{a}_{\mathrm{nuc},s} = -\sum_{n=1}^{N} Z_n \mathbf{T}^{(1)a}_{ns} = Z_a \mathbf{T}^{(2)}_{as}. \tag{2.164}$$

Analogously, the electronic contribution uses the integral derivatives from eq (2.159) and yields

$$\mathcal{F}^{a_l}_{\mathrm{el},s} = -\sum_{pq} t^{(1)a_l}_{pq}(\mathbf{R}_s) D_{pq}. \tag{2.165}$$

Thus, implementation of PE-SCF gradients is straightforward, because for the classical part, only T-tensors of higher order are required. On the quantum side, the electrostatic interaction integrals and field integrals need to be differentiated with respect to nuclear coordinates. To avoid a collapse of the quantum region into the environment through electrostatic attraction, a Lennard-Jones potential can be introduced, for example.[32,59] Gradients for correlated ground and excited state methods will be derived in Section 3.4.

Chapter 3

Polarizable Embedding Combined with ADC

In this chapter, I present my theoretical derivations to combine the PE model with ADC for simulation of various properties. First, I introduce a computationally inexpensive approach to electronic excitation energies based on a self-consistent PE-HF ground state and perturbative corrections.[42,61] Second, I improve this combined scheme through a linear response formalism for PE-ADC, which includes a coupling term of the polarizable environment in the ADC matrix. Through this computationally affordable post-SCF coupling scheme, it is possible for the first time to accurately model higher-order molecular response properties with a polarizable model and ADC. Furthermore, I show the first comprehensive derivation of PE-MP and PE-ADC analytic nuclear gradients, including all orbital response contributions and explicit PE terms required for a successful implementation. This chapter does not only show all the theoretical derivations I performed for PE-ADC, but serves as a blueprint for related solvent models, e.g., polarizable continuum models. The transferability of the response and gradient equations can, in future work, be exploited to derive and implement a plethora of combined methods within the ADC framework. Test calculations and applications of all combined PE-ADC approaches are shown in Chapter 5.

Parts of this chapter have already been published in:

- M. Scheurer, M. F. Herbst, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted, "Polarizable Embedding Combined with the Algebraic Diagrammatic Construction: Tackling Excited States in Biomolecular Systems", *J. Chem. Theory Comput.* **2018**, *14 (9)*, 4870-4883. (Reference 61)
- M. Scheurer, "Polarizable Embedding for the Algebraic-Diagrammatic Construction Scheme", *Springer Fachmedien Wiesbaden*, **2020**. (Reference 42)

3.1 Perturbative Treatment of Excitation Energies

In the simplest possible approach for any post-SCF and excited state method, the self-consistently determined orbitals are kept unchanged. This scheme is called perturbation in the energy (PTE),[62] and is easy to implement because PE contributions only explicitly enter the Fock operator as outlined in Section 2.5.2. Furthermore, the environment contributions do not add to the possibly high computational cost of the correlation treatment. It was shown that the PTE scheme is "most correct" through second order in perturbation theory,[63] and thus applicable, e.g., for a PTE-PE-MP2 treatment. In this case, no PE contribution is included in the computation of the correlation amplitudes other than through the "solvated" orbitals of the PE-HF ground state.

To compensate for the error of not including the environment response dynamically in the excited state treatment or relaxing the environment with respect to a specific excited state (perturbation in the energy and density, PTED scheme),[62] *a posteriori* perturbative energy corrections can be computed for each excited state. The debate over which specific correction terms are needed is not yet finished. In general, there are two commonly used perturbative correction terms: i) the perturbative state-specific (ptSS) correction, which accounts for the change in induction interaction energy upon excitation, and ii) the perturbative linear-response (ptLR) correction, which describes the interaction energy between the oscillating electron density and the polarizable environment. More detailed explanations of these terms will follow. Note that these terms are not unique to atomistic polarizable models, but also occur in polarizable continuum models, i.e., always when mutual polarization effects are considered. In purely electrostatic embedding schemes, none of these corrections are required in the first place, because the environment polarizes the quantum region through a static multipole distribution entering a post-SCF treatment by means of the modified Fock operator. This is an important starting point for deriving the perturbative correction terms for polarizable models.

As stated above, some implementations only make use of the ptSS correction, e.g., EFP combined with ADC and EOM-CCSD,[33,64] whereas it is common in other approaches to treat the linear response of the environment dynamically.[14,27,28,30,32,65] The linear response approach for PE-ADC will be shown in Section 3.2. A comprehensive and thorough analysis carried out by Schwabe[58] using long-range perturbation theory shows that both state-specific and linear response contributions should be included in order to capture as much as possible of the perichromatic shift. The detailed derivation for the following equations are available in Ref. 58, where the same starting point for multiple fragments as in the PE derivation shown in Section 2.5.1 is used. From the perturbation analysis through second order, one obtains four terms that describe the perichromatic shift

$$\Delta E_{\text{shift}}^{0\to n} = \underbrace{\Delta E_{\text{es}}^{0\to n}}_{\text{(i)}} + \underbrace{\Delta E_{\text{ind}}^{0\to n}}_{\text{(ii)}} + \underbrace{\Delta E_{\text{excoupl}}^{0\to n}}_{\text{(iii)}} + \underbrace{\Delta E_{\text{disp}}^{0\to n}}_{\text{(iv)}} . \tag{3.1}$$

The first term (i) denotes the difference in Coulomb interaction energy with the electrostatic environment. This contribution is already accounted for through the HF reference state used in ADC. The mutual change in induction energy (ii) through excitation, however, is only treated in zeroth order. The third term (iii) to consider denotes the non-resonant excitonic coupling. The London dispersion energy (iv) is here neglected, because its contribution is small and not part of the PE formulation anyways.[14,15,58,66] Considering the change in induction energy (ii), one can identify the terms that already entered the ADC procedure in a static manner through the modified Fock operator by construction as

$$\Delta E_{\text{ind}}^{(0),0\to n} = \Delta \boldsymbol{\mathcal{F}}_{\text{ind}}^{0\to n} \boldsymbol{\mu}_{\text{ind}}^{0} = -\boldsymbol{\mathcal{F}}_{\text{el}}^{n} \boldsymbol{\mu}_{\text{ind,el}}^{0} + \boldsymbol{\mathcal{F}}_{\text{el}}^{0} \boldsymbol{\mu}_{\text{ind,el}}^{0}, \tag{3.2}$$

that is, the difference of the electric fields produced by the electrons multiplied with the electronic contribution to induced dipole moments of the HF reference state $\boldsymbol{\mu}_{\text{ind,el}}^{0}$. Note that the electric fields through nuclei of the quantum region and static multipoles of the environment remain unchanged. The zeroth-order perichromatic shift including the above term and the electrostatic contribution from the environment is

$$\Delta E_{\text{shift}}^{(0),0\to n} = \Delta E_{\text{es}}^{0\to n} + \Delta E_{\text{ind}}^{(0),0\to n}. \tag{3.3}$$

Again, for a purely electrostatic embedding scheme, the second term would vanish, and the derivation would be finished at this point, with all the necessary contributions being included by construction. To improve upon this zeroth-order scheme, the dipole moments induced by the excited state wave function need to be incorporated by means of the ptSS correction.[16,66] The electric field $\boldsymbol{\mathcal{F}}_{\text{el}}^{n}$ created by the density of the n-th excited state enters the total induction energy

$$E_{\text{ind}}^{n} = -\frac{1}{2} \boldsymbol{\mathcal{F}}_{\text{el}}^{n} \boldsymbol{\mu}_{\text{ind,el}}^{n}, \tag{3.4}$$

analogously to the ground state expression in eq (2.137). Hence, the difference of (electronic) induction energies between the ground state and excited state n reads

$$\Delta E_{\mathrm{ind}}^{0\to n} = -\frac{1}{2}\left(\boldsymbol{\mathcal{F}}_{\mathrm{el}}^{n}\boldsymbol{\mu}_{\mathrm{ind,el}}^{n} - \boldsymbol{\mathcal{F}}_{\mathrm{el}}^{0}\boldsymbol{\mu}_{\mathrm{ind,el}}^{0}\right). \tag{3.5}$$

Subtracting eq (3.5) from eq (3.2), one obtains the ptSS correction term as

$$\Delta E_{\mathrm{ptSS}}^{0\to n} = -\frac{1}{2}\boldsymbol{\mathcal{F}}_{\mathrm{el}}[\Delta\boldsymbol{\rho}^{n}]\boldsymbol{\mu}_{\mathrm{ind,el}}[\Delta\boldsymbol{\rho}^{n}]. \tag{3.6}$$

To evaluate the above term, the induced dipole equations must be solved with the electrons' electric fields created by the difference density matrix, i.e., the same density-driven approach used for the PE-SCF ground state can be applied. The remaining non-resonant excitonic coupling term (iii) describes the interaction of the transition density $\boldsymbol{\rho}^{n0}$ with the polarizable environment, and is usually dynamically taken into account in response theory approaches.[16] The physical phenomenon formulated in an ad hoc manner yields the ptLR correction term

$$\Delta E_{\mathrm{ptLR}}^{0\to n} = -\boldsymbol{\mathcal{F}}_{\mathrm{el}}[\boldsymbol{\rho}^{n0}]\boldsymbol{\mu}_{\mathrm{ind}}[\boldsymbol{\rho}^{n0}]. \tag{3.7}$$

Again, the correction term is easily evaluated in a density-driven framework by creating the transition electric field and solving the induced dipole moments accordingly. In total, the perturbative correction for the excitation energy of the n-th excited state is

$$\Delta E_{\mathrm{pt}}^{0\to n} = \Delta E_{\mathrm{ptSS}}^{0\to n} + \Delta E_{\mathrm{ptLR}}^{0\to n}. \tag{3.8}$$

In summary, the perturbative corrections for excitation energies entail several advantages:

- The approach is density-driven, i.e., no special routines need to be implemented other than those already needed for PE-SCF.
- Corrections can be obtained for each excited state method for which transition and/or excited state difference densities are available. The above derivation is independent of the electronic structure method.
- An implementation is possible without detailed knowledge about the electronic structure method, i.e., no working equations etc. need to be modified, and only access to the densities is necessary.
- The computational cost is almost zero, because transition and state densities are usually evaluated anyways to obtain transition and excited state dipole moments.

- It is straightforward to judge whether the perturbative treatment fails in the case of the magnitudes of the corrections becoming too large (i.e., the perturbation is not considered “small” anymore).

A small benchmark for excitation energies of solvated molecules and biomolecular applications of the pt-PE-ADC approach is presented in Chapter 5. Because of its simplicity, however, the scheme has some shortcomings:

- Only excitation energies are corrected, but not the excitation vector itself, such that transition and excited state properties are only correct in zeroth order.
- The pole structure of the underlying propagator is not correct anymore, i.e., the corrected excitation energies are not identical to the poles of the response function, but the zeroth-order excitation energies.
- There is no dynamic environment contribution to higher-order response properties, just through the PE-HF reference state.

Motivated by this critique, a linear response formulation of PE-ADC, where the coupling of the transition density to the environment is dynamically included in the ADC matrix, is presented in the following section.

3.2 Linear Response Formalism for PE-ADC

To include a dynamic reaction field directly into the ADC secular equation instead of only perturbative *a posteriori* contributions, the ADC matrix must include coupling terms through linear response theory. Such combinations have recently been published for ADC using continuum solvation models[65,67] and also the PE model presented in the following.[32] The latter approach is based on a PE-CC2 approach,[28] closely related to the first combination of PE and a correlated wave function method, namely PE-CCSD.[27] The same approach was also applied for the second-order polarization propagator (SOPPA).[30] The aforementioned methodologies were not only employed for the computation of electronic excitations, but also for higher-order response properties, such as TPA.

A key aim is to retain the computational efficiency of the already expensive *ab initio* methods, and it would be counterproductive if coupling to a polarizable environment largely increased the cost of the calculation. For this reason, approximate coupling densities for the environment are typically used,[28,30,32] that is, the coupling density is evaluated at a lower level of theory than the secular equations. In the PE-CC2 case, a CCS-like coupling density was used,[28] corresponding to a CIS-like coupling density for ADC.[32]

Evaluating a higher order transition density, i.e., an ISR(2) density in case of ADC(2), in each iteration for each new trial vector would roughly double the computational cost compared to the vacuum case. Hence, the terminology "CIS-like coupling density"[32] means that the ISR(0) transition density (2.57) is used to create the transition electric fields in the environment[34]

$$\rho_{ai}^{(0)} = v_{ia}, \tag{3.9}$$

where only the `vo` block of the density is non-zero and identical to the transposed `ph` block of the excitation vector. The excitation vector, however, has arbitrary order N, resulting in an ADC(N/0) scheme for the coupling. The modified ADC matrix $\bar{\mathbf{M}}$ is a sum of the vacuum matrix (2.52) and the additional PE contribution

$$\bar{\mathbf{M}} = \mathbf{M} + \mathbf{M}^{\mathrm{PE}}. \tag{3.10}$$

The PE contribution to the matrix only has non-zero elements in the `ph-ph` block due to the ISR(0) coupling density, which evaluate to

$$M_{ia,jb}^{\mathrm{PE}} = V_{iajb}^{\mathrm{ind,el}}, \tag{3.11}$$

that is, the induction operator from eq (2.153) transformed to the `ov` space in both electric field operators discarding non-electronic contributions[65]

$$V_{pqrs}^{\mathrm{ind,el}} = -\boldsymbol{\mathcal{F}}_{pq}\mathbf{B}^{-1}\boldsymbol{\mathcal{F}}_{rs}. \tag{3.12}$$

The four-index $V_{iajb}^{\mathrm{ind,el}}$ tensor does not need to be explicitly constructed, because only the matrix-vector product with a trial vector v_{jb} is required

$$r_{ia}^{\mathrm{PE}} = \sum_{jb} V_{iajb}^{\mathrm{ind,el}} v_{jb}. \tag{3.13}$$

Now, the contraction over jb brings us back to the density-driven approach with which *all* PE contributions can be evaluated. In principle, one *could* of course construct the $V_{iajb}^{\mathrm{ind,el}}$ tensor, however, this would overcomplicate the implementation significantly. Using the density-driven approach, the iterative coupling to the environment is achieved as easily as the evaluation of, e.g., perturbative corrections described in the previous section. The PE routine to evaluate operator and energy terms is thus independent of the electronic structure method, as it just needs an "input" density and the information whether only electronic contributions are required. The corresponding implementation aspects are detailed in Section 4.1. The trial vector is used as a density matrix according to eq (3.9) producing transition electric fields

$$\mathcal{F}_a(\mathbf{R}_s)[\boldsymbol{\rho}^{(0)}] = \sum_{jb} \rho_{bj}^{(0)} t_{a,jb}(\mathbf{R}_s). \tag{3.14}$$

Here, the contraction over jb corresponds to evaluation of the electric field expectation value using the ISR(0) or CIS-like coupling density. Consequently, one obtains "transition-induced" moments through

$$\mathbf{B}\boldsymbol{\mu}_{\text{ind,el}}[\boldsymbol{\rho}^{(0)}] = \boldsymbol{\mathcal{F}}_{\text{el}}[\boldsymbol{\rho}^{(0)}]. \tag{3.15}$$

The induction operator formed using only electronic contributions from eq (2.145) is transformed to the `ov` space and directly yields the contribution of $\mathbf{M}^{\text{PE}}$ to the matrix-vector product

$$r_{ia}^{\text{PE}} = V_{ia}^{\text{ind,el}}[\boldsymbol{\rho}^{(0)}]. \tag{3.16}$$

The work by Hättig and co-workers[32] uses exactly the same coupling terms. Due to the structure of their implementation based on CC2,[28] they introduce left and right matrix-vector products, which are of course identical in the case of the Hermitian ADC matrix.[31,35] Finally, the modified LR-PE-ADC eigenvalue problem

$$\bar{\mathbf{M}}\mathbf{Y} = \mathbf{Y}\boldsymbol{\Omega} \tag{3.17}$$

can be solved iteratively as in the vacuum case. With the LR-PE-ADC scheme, however, the environment response directly contributes to the excitation energies and excitation vectors. The physical effect that was previously treated by means of the ptLR correction is, in LR-PE-ADC, included dynamically through the modified ADC matrix. Note that the change in induction energy, i.e., the interaction of the excited state wave function with the environment, is not included using LR. One can still compute a ptSS correction on top of LR-PE-ADC to obtain more accurate perichromatic shifts, especially when large changes in dipole moments occur upon excitation. This combination of LR and a ptSS energy correction was called "corrected PE" (cPE) for PE-TDDFT,[66] first introduced by Caricato et al. for continuum models.[68]

3.3 Molecular Response Properties in the PE-ADC/ISR Framework

The PE model can not only been employed for linear spectroscopies, i.e., excitation energies and transition moments, but also for modeling higher-order spectroscopies that rely on the computation of non-linear molecular response properties. The convenient approach to such properties using the ADC/ISR framework was described in Section 2.4. Now, incorporating the dynamic response of the polarizable environment in an ADC/ISR response function of *any* kind is straightforward having the modified secular matrix (3.10) at hand: the vacuum ADC matrix $\mathbf{M}$ is just replaced by $\bar{\mathbf{M}}$, such that the matrix inversion carried out includes the PE response contributions. Hence, no further modifications or additional terms through the polarizable environment arise, not even for higher-order response functions. This is due to the ISR approach which approximately resolves the SOS in terms of expectation values. Considering the TPA matrix elements in a PE-ADC response scheme, we have

$$S_{AB}^{f,\mathrm{PE}} = \mathbf{F}^{\dagger}(\hat{\mu}_A)\left(\bar{\mathbf{M}} - \frac{\omega_f}{2}\right)^{-1}\mathbf{B}(\hat{\mu}_B)\mathbf{y}_f + \mathbf{F}^{\dagger}(\hat{\mu}_B)\left(\bar{\mathbf{M}} - \frac{\omega_f}{2}\right)^{-1}\mathbf{B}(\hat{\mu}_A)\mathbf{y}_f, \tag{3.18}$$

which only differs from eq (2.86) by the modified PE-ADC secular matrix. Note that also the excitation energy of the final state ω_f first needs to be obtained by solving the LR-PE-ADC eigenvalue problem from eq (3.17). The same replacement procedure can be applied to all other response properties in the ADC/ISR framework as well. Once the modified PE-ADC matrix is in place, no further theoretical derivations are needed, which is very simple and convenient. The simplicity comes from the fact that response functions are recast to the ADC/ISR form by means of expectation values. An alternative approach is to implement response functions through derivatives of the energy.[2] The difference between derivative and expectation value based properties is analyzed in great detail in the excellent work by Hodecker et al.[69] For TPA, the PE model has already been combined with PE-CCSD and PERI-CC2[27–29] using a derivative-based scheme. In this case, there are additional contributions arising through the coupling with the polarizable environment, which makes derivation and implementation more complicated than in the simple ISR scheme for ADC outlined above. Often, there are also approaches where no coupling beyond the polarized ground state is incorporated,[70,71] which is of course much easier from an implementation perspective. A small benchmark of different reaction field schemes within the PE-ADC/ISR framework for TPA strengths will be shown in Chapter 5. To conclude, using the ADC/ISR formalism, it is rather simple and straightforward to include dynamic response of a polarizable environment when modeling all sorts of molecular response properties.

3.4 Analytic Nuclear Gradients

Analytical evaluation of energy gradients with respect to nuclear displacement is one of the cornerstones of electronic structure theory and quantum chemistry in general.[72] According to the Hellmann-Feynman theorem,[73,74] the total derivative of the energy $E = \langle\Psi|\hat{H}|\Psi\rangle$ with respect to an external perturbation ξ is equal to the expectation value of the partial derivative of the Hamiltonian with respect to the same perturbation. The Hellmann-Feynman theorem holds for *exact* wave functions, for wave functions with non-variational parameters Δ, additional terms arise via the chain rule

$$\frac{\mathrm{d}E}{\mathrm{d}\xi} = \langle\Psi|\frac{\partial\hat{H}}{\partial\xi}|\Psi\rangle + \frac{\partial\,\langle\Psi|\hat{H}|\Psi\rangle}{\partial\Delta}\frac{\mathrm{d}\Delta}{\mathrm{d}\xi}. \tag{3.19}$$

An approach to circumvent the explicit and very tedious evaluation of the non-Hellmann-Feynman terms is the so-called *Lagrangian* formalism.[72,75] A Lagrangian energy functional is introduced

$$\mathcal{L} = E + \sum_i \kappa_i f_i(\Delta) \tag{3.20}$$

with the (unknown) Lagrange multipliers $\{\kappa_i\}$. The conditions $\{f_i(\Delta) = 0\}$ ensure that the Lagrangian is stationary with respect to *all* multipliers. An additional requirement is the stationarity of the Lagrangian with respect to all non-variational parameters $\frac{\partial\mathcal{L}}{\partial\Delta} = 0$ which allows one to analytically determine the unknown multipliers and making the Lagrangian functional fulfill the Hellmann-Feynman theorem. Hence, the total derivative of the energy becomes equal to the partial derivative of the Lagrangian

$$\frac{\mathrm{d}E}{\mathrm{d}\xi} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\xi} \overset{!}{=} \frac{\partial\mathcal{L}}{\partial\xi}. \tag{3.21}$$

With the Lagrangian technique, analytical excited state gradients for ADC through third order in perturbation theory were reported by Rehn and Dreuw in 2019.[76] Here, I expand upon their work on *in vacuo* gradients by deriving analytical gradients for combinations of MP and ADC with the PE model. Analytic gradients for PE-MP2 and PE-ADC(2) were first introduced by Hättig and co-workers in 2018,[32] however, the explicit equations given in their work are rather scarce. For this reason, I carry out a comprehensive derivation of all required working equations in the following.

Construction of a General Lagrangian The total derivative of the energy can, for any post-HF method, be constructed through effective one- and two-particle density matrices and an additional set of Lagrange multipliers $\mathbf{\Omega} = \{\omega_{pq}\}$ that need to be contracted with derivatives of the overlap matrix S^{ξ}_{pq}.[77] Let $\boldsymbol{\gamma}^e$ and $\mathbf{\Gamma}^e$ be the effective one- and two-particle density matrix, respectively, the total derivative of the energy is expressed as

$$\frac{\mathrm{d}E}{\mathrm{d}\xi} = \sum_{pq} h^{\xi}_{pq}\gamma^{e}_{pq} + \frac{1}{4}\sum_{pqrs} \langle pq||rs\rangle^{\xi}\,\Gamma^{e}_{pqrs} + \sum_{pq} S^{\xi}_{pq}\omega_{pq}, \tag{3.22}$$

with the derivative of the core Hamiltonian h^{ξ}_{pq} and the anti-symmetrized two-electron integrals $\langle pq||rs\rangle^{\xi}$. Both the effective densities and the $\mathbf{\Omega}$ multipliers are obtained via the Lagrangian formalism. For a canonical HF reference state, the Fock operator is diagonal, $f_{pq} - \delta_{pq}\varepsilon_p = 0$, and the overlap matrix is equal to the unity matrix, $S_{pq} - \delta_{pq} = 0$ (see eq (2.35)). To incorporate stationary conditions for the parameters of the correlation treatment $\mathbf{T} = \{t_z\}$, the equations $\{f_z(t_z)\}$ that satisfy the condition $f_z(t_z) = 0$ are defined. Now, with all three conditions at hand, a general Lagrangian is formed[76,78]

$$\mathcal{L}\left(\mathbf{C}, \mathbf{T}, \mathbf{\Lambda}, \mathbf{\Omega}, \bar{\mathbf{T}}\right) = E(\mathbf{C}, \mathbf{T}) + \frac{1}{2}\sum_{pq} \lambda_{pq}(f_{pq} - \delta_{pq}\varepsilon_p) + \sum_{pq} \omega_{pq}(S_{pq} - \delta_{pq}) + \sum_{z} \bar{t}_z f_z(t_z). \tag{3.23}$$

All terms except the energy $E(\mathbf{C}, \mathbf{T})$ of the correlated method in the Lagrangian are equal to zero, and the derivatives of the Lagrangian with respect to all multipliers vanish as well. Note that the $\mathbf{\Omega}$ and $\mathbf{\Lambda}$ multiplier matrices are symmetric. Through the following requirements

$$\frac{\partial \mathcal{L}}{\partial \mathbf{C}} \overset{!}{=} 0 \tag{3.24}$$

and

$$\frac{\partial \mathcal{L}}{\partial \mathbf{T}} \overset{!}{=} 0, \tag{3.25}$$

it is possible to determine all unknown multipliers in the Lagrangian. Once this is achieved, the partial derivative of the Lagrangian evaluates as

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \xi} =& \sum_{pq} h_{pq}^{\xi} \underbrace{\left(\gamma_{pq}(\mathbf{T}) + \gamma_{pq}^{O}(\mathbf{\Lambda}) + \gamma_{pq}^{A}(\bar{\mathbf{T}})\right)}_{\gamma_{pq}^{e}} \\ &+ \frac{1}{4} \sum_{pqrs} \langle pq||rs\rangle^{\xi} \underbrace{\left(\Gamma_{pqrs}(\mathbf{T}) + \Gamma_{pqrs}^{O}(\mathbf{\Lambda}) + \Gamma_{pqrs}^{A}(\bar{\mathbf{T}})\right)}_{\Gamma_{pqrs}^{e}} + \sum_{pq} S_{pq}^{\xi} \omega_{pq}.\end{aligned} \tag{3.26}$$

Here, the effective density matrices are split up into different contributions. First, the so-called *unrelaxed* densities $\boldsymbol{\gamma}(\mathbf{T})$ and $\mathbf{\Gamma}(\mathbf{T})$ depend on the parameters of the correlation treatment, but not on the Lagrange multipliers. Second, the orbital response contributions through $\boldsymbol{\gamma}^{O}(\mathbf{\Lambda})$ and $\mathbf{\Gamma}^{O}(\mathbf{\Lambda})$ take into account stationarity with respect to variations in the orbital basis and depend on the $\mathbf{\Lambda}$ multipliers. The third contribution consists of the amplitude-relaxed densities $\boldsymbol{\gamma}^{A}(\bar{\mathbf{T}})$ and $\mathbf{\Gamma}^{A}(\bar{\mathbf{T}})$ ensuring that the Lagrangian is stationary with respect to parameters of the correlation method. The orbital response contributions are independent of the underlying method, whereas the amplitude-relaxed densities are method-dependent. Turning the focus back on PE within a Lagrangian formalism for gradients, we notice that there is no explicit PE contribution to the correlation treatment. Thus, all equations to determine unrelaxed and amplitude-relaxed densities are unaffected by inclusion of PE and are identical to the vacuum form. The orbital response contributions, however, will contain terms arising from PE through the modified Fock operator in the PTE case, and in addition a modified energy expression for the LR-PE-ADC case. Derivation of the orbital response contributions closely follows the notation from Levchenko et al.[78] and Rehn et al.[76]. Some details are omitted for brevity and can be found in the original literature.

3.4.1 Orbital Response Contributions via the PE Fock Operator

The following derivation for orbital response expressions can be used for all PTE schemes using the PE model, i.e., no explicit PE contributions enter the correlation treatment except for the modified Fock operator and the modified energy of the reference state. This applies, e.g., to PTE-PE-MP2 and PTE-PE-ADC(N). Defining the energy in eq (3.23) in terms of the Fock matrix and density matrices, a “re-sorted” Lagrangian is found as[76,78]

$$\mathcal{L} = \sum_{pq} f_{pq} \gamma'_{pq} + \frac{1}{4} \sum_{pqrs} \langle pq||rs\rangle \, \Gamma'_{pqrs} + \frac{1}{2} \sum_{pq} \lambda_{pq} (f_{pq} - \delta_{pq} \varepsilon_p) + \sum_{pq} \omega_{pq} (S_{pq} - \delta_{pq}). \tag{3.27}$$

In the equation above, the density matrices collect everything except the orbital response part, that is[76]

$$\boldsymbol{\gamma}' \equiv \boldsymbol{\gamma}(\mathbf{T}) + \boldsymbol{\gamma}^A(\bar{\mathbf{T}}) \quad \text{and} \quad \boldsymbol{\Gamma}' \equiv \tilde{\boldsymbol{\Gamma}}(\mathbf{T}) + \boldsymbol{\Gamma}^A(\bar{\mathbf{T}}) \tag{3.28}$$

with the non-separable part of the two-particle density matrix $\tilde{\boldsymbol{\Gamma}}$.[76] As follows from eq (3.24), we now need to differentiate the above Lagrangian functional and avoid partially transformed tensors[78] via

$$\sum_\mu C_{\mu u} \frac{\partial \mathcal{L}}{\partial C_{\mu t}} \overset{!}{=} 0, \tag{3.29}$$

which requires derivatives of the Fock operator, the two-electron integrals, and the overlap matrix with respect to orbital coefficients. Their evaluation is straightforward for the vacuum case.[76,78] The PE contributions, require further attention though. Using the definition of the PE Fock operator (2.152), the matrix elements of the composite Fock operator are given by

$$f_{pq}^{\mathrm{PE}} = h_{pq} + \sum_i \langle pi||qi\rangle + V_{pq}^{\mathrm{es}} + \sum_i V_{pqii}^{\mathrm{ind}}. \tag{3.30}$$

As stated above, the derivatives of the first two terms are identical to the vacuum case. The derivative of the electrostatic PE operator is simply

$$\sum_\mu C_{\mu u} \frac{\partial V_{pq}^{\mathrm{es}}}{\partial C_{\mu t}} = V_{uq}^{\mathrm{es}} \delta_{pt} + V_{pu}^{\mathrm{es}} \delta_{qt}. \tag{3.31}$$

For the derivative of the induction operator defined in eq (2.153), it is convenient to carry out the following split-up

$$V_{pq}^{\mathrm{ind}} = \sum_i \underbrace{-\boldsymbol{\mathcal{F}}_{pq} \mathbf{B}^{-1} \boldsymbol{\mathcal{F}}_{ii}}_{V_{pqii}^{\mathrm{ind,el}}} - \boldsymbol{\mathcal{F}}_{pq} \underbrace{\mathbf{B}^{-1} \left(\boldsymbol{\mathcal{F}}_{\mathrm{nuc}} + \boldsymbol{\mathcal{F}}_{\mathrm{mul}} \right)}_{\boldsymbol{\mu}_{\mathrm{ind,ne}}}. \tag{3.32}$$

The second term only depends on two general orbital indices like the electrostatic PE operator, and consequently the derivative is trivial

$$\sum_\mu C_{\mu u} \frac{\partial \boldsymbol{\mathcal{F}}_{pq} \boldsymbol{\mu}_{\mathrm{ind,ne}}}{\partial C_{\mu t}} = \delta_{pt} \boldsymbol{\mathcal{F}}_{uq} \boldsymbol{\mu}_{\mathrm{ind,ne}} + \delta_{qt} \boldsymbol{\mathcal{F}}_{pu} \boldsymbol{\mu}_{\mathrm{ind,ne}}. \tag{3.33}$$

The derivative of the 'electronic' part of the induction operator with respect to MO coefficients is

$$\sum_{\mu} C_{\mu u} \sum_{i} \frac{\partial V^{\text{ind,el}}_{pqii}}{\partial C_{\mu t}} = \sum_{\mu} C_{\mu u} \sum_{i} \frac{\partial C_{\gamma p} C_{\nu q} V^{\text{ind,el}}_{\gamma\nu\delta\sigma} C_{\delta i} C_{\sigma i}}{\partial C_{\mu t}} \tag{3.34}$$

$$= \delta_{pt} V^{\text{ind,el}}_{uqii} + \delta_{qt} V^{\text{ind,el}}_{puii} + \delta_{it} V^{\text{ind,el}}_{pqui} + \delta_{it} V^{\text{ind,el}}_{pqiu}. \tag{3.35}$$

The first two terms and the derivatives in eqs (3.31) and (3.33) above share the same structure due to the Kronecker deltas, one realizes that these terms are nothing but elements of the Fock operator itself

$$\begin{aligned} V^{\text{PE}}_{uq} \delta_{pt} + V^{\text{PE}}_{pu} \delta_{qt} &= V^{\text{es}}_{uq} \delta_{pt} + V^{\text{es}}_{pu} \delta_{qt} \\ &\quad - \delta_{pt} \boldsymbol{\mathcal{F}}_{uq} \boldsymbol{\mu}_{\text{ind,ne}} - \delta_{qt} \boldsymbol{\mathcal{F}}_{pu} \boldsymbol{\mu}_{\text{ind,ne}} \\ &\quad + \delta_{pt} V^{\text{ind,el}}_{uqii} + \delta_{qt} V^{\text{ind,el}}_{puii} \end{aligned} \tag{3.36}$$

$$= (V^{\text{es}}_{uq} + V^{\text{ind,el}}_{uq}) \delta_{pt} + (V^{\text{es}}_{pu} + V^{\text{ind,el}}_{pu}) \delta_{qt}. \tag{3.37}$$

Hence, only the electronic part, i.e., the last two terms of eq (3.35), contributes beyond the vacuum case. With this contribution at hand, the derivative of the entire Lagrangian (3.27) with respect to orbital coefficients is[76,78]

$$\begin{aligned} \sum_{\mu} C_{\mu u} \frac{\partial \mathcal{L}}{\partial C_{\mu t}} &= \left(2\gamma'_{ut} + \lambda_{ut}\right) \varepsilon_u \\ &\quad + \sum_{pq} \left(2\gamma'_{pq} + \lambda_{pq}\right) \langle pu || qt \rangle \, \delta_{t \in \text{occ.}} \\ &\quad + \sum_{pq} \left(2\gamma'_{pq} + \lambda_{pq}\right) V^{\text{ind,el}}_{pqut} \delta_{t \in \text{occ.}} \\ &\quad + \sum_{pqr} \Gamma'_{tpqr} \langle up || qr \rangle + 2\omega_{ut}. \end{aligned} \tag{3.38}$$

Swapping indices and subtracting one term from the other, one obtains

$$\begin{aligned}\sum_\mu C_{\mu u}\frac{\partial\mathcal{L}}{\partial C_{\mu t}}-\sum_\mu C_{\mu t}\frac{\partial\mathcal{L}}{\partial C_{\mu u}} = {} & \left(2\gamma''_{ut}+\lambda_{ut}\right)\left(\varepsilon_u-\varepsilon_t\right) \\ & +\sum_{pq}\left(2\gamma''_{pq}+\lambda_{pq}\right)\left(\langle pu||qt\rangle\,\delta_{t\in\text{occ.}}-\langle pt||qu\rangle\,\delta_{u\in\text{occ.}}\right) \\ & +\sum_{pq}\left(2\gamma''_{pq}+\lambda_{pq}\right)\left(V^{\text{ind,el}}_{pqut}\delta_{t\in\text{occ.}}-V^{\text{ind,el}}_{pqtu}\delta_{u\in\text{occ.}}\right) \\ & +\sum_{pqr}\left(\Gamma''_{tpqr}\,\langle up||qr\rangle-\Gamma''_{upqr}\,\langle tp||qr\rangle\right).\end{aligned} \tag{3.39}$$

The HF part of the densities in the equation above cancels out. For this reason, the one- and two-particle densities which lack the HF contribution $\boldsymbol{\gamma}''$ and $\boldsymbol{\Gamma}''$ are used.[76] The Lagrange multipliers for the overlap matrix vanish due to symmetry ($\omega_{ut}-\omega_{tu}=0$). Setting $u=a$ and $t=i$, one finds the equation to determine λ_{ia} as

$$\begin{aligned}\sum_\mu C_{\mu a}\frac{\partial\mathcal{L}}{\partial C_{\mu i}}-\sum_\mu C_{\mu i}\frac{\partial\mathcal{L}}{\partial C_{\mu a}} = {} & \left(2\gamma''_{ia}+\lambda_{ia}\right)\left(\varepsilon_a-\varepsilon_i\right) \\ & +\sum_{pq}\left(2\gamma''_{pq}+\lambda_{pq}\right)\langle pa||qi\rangle \\ & +\sum_{pq}\left(2\gamma''_{pq}+\lambda_{pq}\right)V^{\text{ind,el}}_{pqai} \\ & +\sum_{pqr}\left(\Gamma''_{ipqr}\,\langle ap||qr\rangle-\Gamma''_{apqr}\,\langle ip||qr\rangle\right),\end{aligned} \tag{3.40}$$

which, splitting up summations in canonical blocks, expands to

$$\begin{aligned}0 = {} & \lambda_{ia}\left(\varepsilon_a-\varepsilon_i\right) \\ & +\sum_{kl}(2\gamma''_{kl}+\lambda_{kl})\,\langle ka||li\rangle+\sum_{kl}\left(2\gamma''_{kl}+\lambda_{kl}\right)V^{\text{ind,el}}_{klai} \\ & +\sum_{bc}(2\gamma''_{bc}+\lambda_{bc})\,\langle ba||ci\rangle+\sum_{bc}\left(2\gamma''_{bc}+\lambda_{bc}\right)V^{\text{ind,el}}_{bcai} \\ & +\sum_{jc}(2\gamma''_{jb}+\lambda_{jb})\,\langle ja||bi\rangle+\sum_{jb}\left(2\gamma''_{jb}+\lambda_{jb}\right)V^{\text{ind,el}}_{jbai} \\ & +\sum_{bj}(2\gamma''_{bj}+\lambda_{bj})\,\langle ba||ji\rangle+\sum_{bj}\left(2\gamma''_{bj}+\lambda_{bj}\right)V^{\text{ind,el}}_{bjai} \\ & +\sum_{pqr}\left(\Gamma''_{ipqr}\,\langle ap||qr\rangle-\Gamma''_{apqr}\,\langle ip||qr\rangle\right).\end{aligned} \tag{3.41}$$

Exploiting symmetry and removing zero blocks ($\lambda_{kl} = 0$, $\lambda_{bc} = 0$), one can sort the previous equation into λ-dependent and λ-independent parts

$$\begin{aligned}
0 = {} & \lambda_{ia}\left(\varepsilon_a - \varepsilon_i\right) \\
& + \sum_{jb} \lambda_{jb}\left(\langle ij||ab\rangle - \langle ib||ja\rangle\right) \\
& + \sum_{jb} \lambda_{jb}\left(V^{\text{ind,el}}_{jbai} + V^{\text{ind,el}}_{bjai}\right) \\
& + 2\sum_{kl} \gamma''_{kl}\langle li||ka\rangle + 2\sum_{kl} \gamma''_{kl} V^{\text{ind,el}}_{klai} \\
& + 2\sum_{bc} \gamma''_{bc}\langle ic||ab\rangle + 2\sum_{bc} \gamma''_{bc} V^{\text{ind,el}}_{bcai} \\
& + \sum_{pqr}\left(\Gamma''_{ipqr}\langle ap||qr\rangle - \Gamma''_{apqr}\langle ip||qr\rangle\right) .
\end{aligned} \tag{3.42}$$

Note that PE terms are found in a) the part of the orbital Hessian matrix that multiplies the Lagrange multipliers λ_{jb} (third term), and b) two static contributions to the right-hand side of the linear equations (fifth and seventh term). Moving all λ_{ia}-independent terms on one side, the equation to iteratively determine the multipliers is

$$\lambda_{ia}\left(\varepsilon_a - \varepsilon_i\right) + \sum_{jb} \lambda_{jb}\left(\langle ij||ab\rangle - \langle ib||ja\rangle\right) + \sum_{jb} \lambda_{jb}\left(V^{\text{ind,el}}_{jbai} + V^{\text{ind,el}}_{bjai}\right) = k_{ia} \tag{3.43}$$

with the right-hand side

$$\begin{aligned}
k_{ia} = {} & -2\sum_{kl} \gamma''_{kl}\langle li||ka\rangle - 2\sum_{kl} \gamma''_{kl} V^{\text{ind,el}}_{klai} \\
& - 2\sum_{bc} \gamma''_{bc}\langle ic||ab\rangle - 2\sum_{bc} \gamma''_{bc} V^{\text{ind,el}}_{bcai} \\
& - \sum_{pqr}\left(\Gamma''_{ipqr}\langle ap||qr\rangle - \Gamma''_{apqr}\langle ip||qr\rangle\right) .
\end{aligned} \tag{3.44}$$

Once the multipliers λ_{ia} are iteratively determined, the unknown $\mathbf{\Omega}$ multipliers are accessible. Starting from eq (3.38), the multipliers are given by

$$\begin{aligned}\omega_{ut} =& -\left(\gamma'_{ut} + \frac{1}{2}\lambda_{ut}\right)\varepsilon_u \\ & -\sum_{pq}\left(\gamma'_{pq} + \frac{1}{2}\lambda_{pq}\right)\langle pu||qt\rangle\,\delta_{t\in\text{occ.}} \\ & -\sum_{pq}\left(\gamma'_{pq} + \frac{1}{2}\lambda_{pq}\right)V^{\text{ind,el}}_{pqut}\delta_{t\in\text{occ.}} \\ & -\frac{1}{2}\sum_{pqr}\Gamma'_{tpqr}\langle up||qr\rangle\,. \end{aligned} \tag{3.45}$$

Three different canonical blocks of $\{\omega_{pq}\}$ are obtained, starting with $u = i$ and $t = j$ yields

$$\begin{aligned}\omega_{ij} =& -\gamma'_{ij}\varepsilon_i \\ & -\sum_{pq}\left(\gamma'_{pq} + \frac{1}{2}\lambda_{pq}\right)\langle pi||qj\rangle \\ & -\sum_{pq}\left(\gamma'_{pq} + \frac{1}{2}\lambda_{pq}\right)V^{\text{ind,el}}_{pqij} \\ & -\frac{1}{2}\sum_{pqr}\Gamma'_{jpqr}\langle ip||qr\rangle\,. \end{aligned} \tag{3.46}$$

Next, with $u = a$ and $t = b$, we have

$$\begin{aligned}\omega_{ab} =& -\gamma'_{ab}\varepsilon_a \\ & -\frac{1}{2}\sum_{pqr}\Gamma'_{bpqr}\langle ap||qr\rangle\,. \end{aligned} \tag{3.47}$$

The last block is found for $u = i$ and $t = a$, that is

$$\begin{aligned}\omega_{ia} =& -\left(\gamma'_{ia} + \frac{1}{2}\lambda_{ia}\right)\varepsilon_i \\ & -\frac{1}{2}\sum_{pqr}\Gamma'_{apqr}\langle ip||qr\rangle\,. \end{aligned} \tag{3.48}$$

Hence, only the ω_{ij} multipliers contain explicit PE contributions when a PTE scheme is employed.

3.4.2 Orbital Response for LR Schemes

In Section 3.2, I have presented the LR-PE-ADC scheme, where an additional coupling term to the environment enters the ADC matrix. This goes beyond a PTE treatment, requiring extra contributions to the orbital response equations beyond the modified Fock operator. The total energy of an excited state n with the excitation vector $\mathbf{x} \equiv \mathbf{x}_n$ within LR-PE-ADC is

$$E_n = E^{(0)} + \mathbf{x}^\dagger \mathbf{M} \mathbf{x} + \mathbf{x}^\dagger \mathbf{M}^{\mathrm{PE}} \mathbf{x} = E^{(0)} + \mathbf{x}^\dagger \mathbf{M} \mathbf{x} + \underbrace{\sum_{klcd} x_{kc} x_{ld} V_{kcld}^{\mathrm{ind,el}}}_{\Delta E^{\mathrm{PE}}} \tag{3.49}$$

To obtain the contributions for the $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ multipliers, one requires the derivative of the additional ΔE^{PE} with respect to orbital coefficients

$$\sum_{\mu} C_{\mu u} \sum_{klcd} x_{kc} x_{ld} \frac{\partial V_{kcld}^{\mathrm{ind,el}}}{\partial C_{\mu t}}. \tag{3.50}$$

The derivative expands to

$$\frac{\partial V_{kcld}^{\mathrm{ind,el}}}{\partial C_{\mu t}} = \sum_{\gamma\nu\theta\sigma} \frac{\partial C_{\gamma k} C_{\nu c} C_{\theta l} C_{\sigma d} V_{\gamma\nu\theta\sigma}^{\mathrm{ind,el}}}{\partial C_{\mu t}} \tag{3.51}$$

$$\begin{aligned} &= \sum_{\gamma\nu\theta\sigma} \delta_{\gamma\mu} \delta_{kt} C_{\nu c} C_{\theta l} C_{\sigma d} V_{\gamma\nu\theta\sigma}^{\mathrm{ind,el}} \\ &\quad + \sum_{\gamma\nu\theta\sigma} C_{\gamma k} \delta_{\nu\mu} \delta_{ct} C_{\theta l} C_{\sigma d} V_{\gamma\nu\theta\sigma}^{\mathrm{ind,el}} \\ &\quad + \sum_{\gamma\nu\theta\sigma} C_{\gamma k} C_{\nu c} \delta_{\theta\mu} \delta_{lt} C_{\sigma d} V_{\gamma\nu\theta\sigma}^{\mathrm{ind,el}} \\ &\quad + \sum_{\gamma\nu\theta\sigma} C_{\gamma k} C_{\nu c} C_{\theta l} \delta_{\sigma\mu} \delta_{dt} V_{\gamma\nu\theta\sigma}^{\mathrm{ind,el}} \end{aligned} \tag{3.52}$$

$$\begin{aligned} &= \delta_{kt} \sum_{\nu\theta\sigma} C_{\nu c} C_{\theta l} C_{\sigma d} V_{\mu\nu\theta\sigma}^{\mathrm{ind,el}} \\ &\quad + \delta_{ct} \sum_{\gamma\theta\sigma} C_{\gamma k} C_{\theta l} C_{\sigma d} V_{\gamma\mu\theta\sigma}^{\mathrm{ind,el}} \\ &\quad + \delta_{lt} \sum_{\gamma\nu\sigma} C_{\gamma k} C_{\nu c} C_{\sigma d} V_{\gamma\nu\mu\sigma}^{\mathrm{ind,el}} \\ &\quad + \delta_{dt} \sum_{\gamma\nu\theta} C_{\gamma k} C_{\nu c} C_{\theta l} V_{\gamma\nu\theta\mu}^{\mathrm{ind,el}} \end{aligned} \tag{3.53}$$

$$= \delta_{kt} V_{\mu cld}^{\mathrm{ind,el}} + \delta_{ct} V_{k\mu ld}^{\mathrm{ind,el}} + \delta_{lt} V_{kc\mu d}^{\mathrm{ind,el}} + \delta_{dt} V_{kcl\mu}^{\mathrm{ind,el}}. \tag{3.54}$$

Using this result, we have

$$\sum_\mu C_{\mu u} \frac{\partial V^{\text{ind,el}}_{kcld}}{\partial C_{\mu t}} = \delta_{kt} V^{\text{ind,el}}_{ucld} + \delta_{ct} V^{\text{ind,el}}_{kuld} + \delta_{lt} V^{\text{ind,el}}_{kcud} + \delta_{dt} V^{\text{ind,el}}_{kclu}, \tag{3.55}$$

and consequently define

$$\begin{aligned} b_{ut} = & \sum_{klcd} x_{kc} x_{ld} \delta_{kt} V^{\text{ind,el}}_{ucld} \\ & + \sum_{klcd} x_{kc} x_{ld} \delta_{ct} V^{\text{ind,el}}_{kuld} \\ & + \sum_{klcd} x_{kc} x_{ld} \delta_{lt} V^{\text{ind,el}}_{kcud} \\ & + \sum_{klcd} x_{kc} x_{ld} \delta_{dt} V^{\text{ind,el}}_{kclu} \end{aligned} \tag{3.56}$$

$$\begin{aligned} = & \sum_{lcd} x_{tc} x_{ld} V^{\text{ind,el}}_{ucld} \delta_{t\in\text{occ.}} + \sum_{kld} x_{kt} x_{ld} V^{\text{ind,el}}_{kuld} \delta_{t\in\text{virt.}} \\ & + \sum_{kcd} x_{kc} x_{td} V^{\text{ind,el}}_{kcud} \delta_{t\in\text{occ.}} + \sum_{klc} x_{kc} x_{lt} V^{\text{ind,el}}_{kclu} \delta_{t\in\text{virt.}}. \end{aligned} \tag{3.57}$$

Setting the indices u, t to different orbital spaces, one obtains the contributions for both $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ Lagrange multipliers. To determine $\mathbf{\Lambda}$, we set $u = a$ and $t = i$

$$b_{ai} = 2 \sum_{lcd} x_{ic} x_{ld} V^{\text{ind,el}}_{acld}, \tag{3.58}$$

and also $u = i$ and $t = a$, that is

$$b_{ia} = 2 \sum_{klc} x_{kc} x_{la} V^{\text{ind,el}}_{kcli}. \tag{3.59}$$

The difference $b_{ai} - b_{ia}$ then contributes to the right-hand side of eq (3.42) as a constant term, independent of λ_{ia}. For the contributions to ω_{ij} (eq (3.46)), we set $u = i$ and $t = j$ and find

$$b_{ij} = 2 \sum_{lcd} x_{jc} x_{ld} V^{\text{ind,el}}_{ldic}. \tag{3.60}$$

In the same manner, one obtains the contribution to ω_{ab} (eq (3.47)) via $u = a$ and $t = b$ as

$$b_{ab} = 2\sum_{kld} x_{kb} x_{ld} V^{\text{ind,el}}_{kald}. \tag{3.61}$$

The additional ω_{ia} term (eq (3.48)) is accessible via b_{ia}. In summary, the adapted $\mathbf{\Omega}$ multipliers are

$$\omega'_{ij} = \omega_{ij} - \frac{1}{2} b_{ij} \tag{3.62}$$

$$\omega'_{ab} = \omega_{ab} - \frac{1}{2} b_{ab} \tag{3.63}$$

$$\omega'_{ia} = \omega_{ia} - \frac{1}{2} b_{ia}. \tag{3.64}$$

Table 3.1: Summary of all orbital response working equations through the PE ground state (Fock operator) in PTE, and the matrix term in LR-PE

Term	PTE	LR-PE
λ iteration	$\sum_{jb} \lambda_{jb} V^{\text{ind,el}}_{jbai} + \lambda_{bj} V^{\text{ind,el}}_{bjai}$	-
k_{ia}	$-2\sum_{kl} \gamma''_{kl} V^{\text{ind,el}}_{klai} - 2\sum_{bc} \gamma''_{bc} V^{\text{ind,el}}_{bcai}$	$-2\sum_{lcd} x_{ic} x_{ld} V^{\text{ind,el}}_{acld} + 2\sum_{klc} x_{kc} x_{la} V^{\text{ind,el}}_{kcli}$
ω_{ij}	$-\sum_{pq} \left(\gamma'_{pq} + \frac{1}{2}\lambda_{pq}\right) V^{\text{ind,el}}_{pqij}$	$-\sum_{lcd} x_{jc} x_{ld} V^{\text{ind,el}}_{ldic}$
ω_{ab}	-	$-\sum_{kld} x_{kb} x_{ld} V^{\text{ind,el}}_{kald}$
ω_{ia}	-	$-\sum_{klc} x_{kc} x_{la} V^{\text{ind,el}}_{kcli}$

With these equations at hand, all $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ multipliers are available. No explicit contributions are required for the amplitude response Lagrange multipliers $\bar{\mathbf{T}}$ because no environment-specific terms enter the correlation treatment.

3.4.3 PE Contributions to the PTE-PE Gradient

As a final step to execute the derivative of the Lagrangian with respect to nuclear coordinates, the explicit PE contributions need to be identified. These arise, in a PTE scheme through a) the PE contribution to the HF energy (2.130), and b) through the Fock operator (2.150). All other contributions needed for the derivative, including method-specific density matrices for the amplitude response of ADC schemes can be found in Ref. 76. For convenience, a difference density matrix between the effective density and the HF density is defined as

$$\boldsymbol{\gamma}^{\Delta} = \boldsymbol{\gamma}^{e} - \boldsymbol{\gamma}^{\mathrm{HF}} \tag{3.65}$$

with

$$\gamma_{ij}^{\mathrm{HF}} = \delta_{ij}. \tag{3.66}$$

Now, using this definition, let

$$\Delta\mathcal{L}^{\text{PTE-PE}} = E_{\mathrm{PE}}[\boldsymbol{\gamma}^{\mathrm{HF}}] + \sum_{pq} V_{pq}^{\mathrm{PE}} \gamma_{pq}^{\Delta} \tag{3.67}$$

be the difference to the vacuum Lagrangian. To simplify the expression, one can split up the above terms to

$$\begin{aligned} \Delta\mathcal{L}^{\text{PTE-PE}} &= \sum_{pq} V_{pq}^{\mathrm{es}} \gamma_{pq}^{\mathrm{HF}} + \sum_{pq} \left(V_{pq}^{\mathrm{es}} + V_{pq}^{\mathrm{ind}} \right) \gamma_{pq}^{\Delta} + E_{\mathrm{es}}^{\mathrm{nuc}} + E_{\mathrm{ind}} & (3.68) \\ &= \sum_{pq} V_{pq}^{\mathrm{es}} \left(\gamma_{pq}^{\mathrm{HF}} + \gamma_{pq}^{\Delta} \right) + \sum_{pq} V_{pq}^{\mathrm{ind}} \gamma_{pq}^{\Delta} + E_{\mathrm{es}}^{\mathrm{nuc}} + E_{\mathrm{ind}} & (3.69) \end{aligned}$$

For clarity, let us split up the Lagrangian in an electrostatic part

$$\Delta\mathcal{L}_{\mathrm{es}}^{\text{PTE-PE}} = \sum_{pq} V_{pq}^{\mathrm{es}} \gamma_{pq}^{e} + E_{\mathrm{es}}^{\mathrm{nuc}} \tag{3.70}$$

and an induction part

$$\Delta\mathcal{L}_{\mathrm{ind}}^{\text{PTE-PE}} = \sum_{pq} V_{pq}^{\mathrm{ind}} \gamma_{pq}^{\Delta} + E_{\mathrm{ind}}. \tag{3.71}$$

The derivative of the electrostatic contribution is given by

$$\begin{aligned} \frac{\partial \Delta\mathcal{L}_{\mathrm{es}}^{\text{PTE-PE}}}{\partial \xi} &= \sum_{pq} \frac{\partial V_{pq}^{\mathrm{es}}}{\partial \xi} \gamma_{pq}^{e} + \frac{\partial E_{\mathrm{es}}^{\mathrm{nuc}}}{\partial \xi} & (3.72) \\ &= E_{\mathrm{es}}^{\mathrm{el},\xi}[\boldsymbol{\gamma}^{e}] + E_{\mathrm{es}}^{\mathrm{nuc},\xi} & (3.73) \end{aligned}$$

which can be evaluated via eq (2.158) using the total one-particle density γ^e, and eq (2.157), which is method-independent. The induction part requires more handiwork due to the fact that the induction operator itself is density-dependent and does not directly yield the induction energy. Expanding eq (3.71) yields

$$\Delta\mathcal{L}_{\text{ind}}^{\text{PTE-PE}} = -\frac{1}{2}\boldsymbol{\mu}_{\text{ind}}[\gamma^{\text{HF}}]\boldsymbol{\mathcal{F}}[\gamma^{\text{HF}}] - \boldsymbol{\mu}_{\text{ind}}[\gamma^{\text{HF}}]\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]. \tag{3.74}$$

Now carrying out the partial derivative of the induction Lagrangian, we have that

$$\begin{aligned}\frac{\partial\Delta\mathcal{L}_{\text{ind}}^{\text{PTE-PE}}}{\partial\xi} =& -\boldsymbol{\mu}_{\text{ind}}[\gamma^{\text{HF}}]\frac{\partial\boldsymbol{\mathcal{F}}[\gamma^{\text{HF}}]}{\partial\xi}\\ &-\frac{\partial}{\partial\xi}\left(\boldsymbol{\mu}_{\text{ind}}[\gamma^{\text{HF}}]\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]\right),\end{aligned} \tag{3.75}$$

where we have used eq (2.162) to simplify the derivative of the PE-HF induction energy. The second term in the above equation yields two summands due to the chain rule

$$\begin{aligned}-\frac{\partial}{\partial\xi}\left(\boldsymbol{\mu}_{\text{ind}}[\gamma^{\text{HF}}]\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]\right) =& -\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]}{\partial\xi}\mathbf{B}^{-1}\left(\boldsymbol{\mathcal{F}}_{\text{nuc}} + \boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}] + \boldsymbol{\mathcal{F}}_{\text{mul}}\right)\\ &-\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]\mathbf{B}^{-1}\left(\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}]}{\partial\xi} + \frac{\partial\boldsymbol{\mathcal{F}}_{\text{nuc}}}{\partial\xi}\right).\end{aligned} \tag{3.76}$$

Summing up all terms gives

$$\begin{aligned}\frac{\partial\Delta\mathcal{L}_{\text{ind}}^{\text{PTE-PE}}}{\partial\xi} =& -\left(\boldsymbol{\mathcal{F}}_{\text{nuc}} + \boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}] + \boldsymbol{\mathcal{F}}_{\text{mul}}\right)\mathbf{B}^{-1}\left(\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}]}{\partial\xi} + \frac{\partial\boldsymbol{\mathcal{F}}_{\text{nuc}}}{\partial\xi}\right)\\ &-\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]}{\partial\xi}\mathbf{B}^{-1}\left(\boldsymbol{\mathcal{F}}_{\text{nuc}} + \boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}] + \boldsymbol{\mathcal{F}}_{\text{mul}}\right)\\ &-\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]\mathbf{B}^{-1}\left(\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}]}{\partial\xi} + \frac{\partial\boldsymbol{\mathcal{F}}_{\text{nuc}}}{\partial\xi}\right).\end{aligned} \tag{3.77}$$

$$\begin{aligned}=& -\left(\boldsymbol{\mathcal{F}}_{\text{nuc}} + \boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{e}] + \boldsymbol{\mathcal{F}}_{\text{mul}}\right)\mathbf{B}^{-1}\left(\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}]}{\partial\xi} + \frac{\partial\boldsymbol{\mathcal{F}}_{\text{nuc}}}{\partial\xi}\right)\\ &-\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]}{\partial\xi}\mathbf{B}^{-1}\left(\boldsymbol{\mathcal{F}}_{\text{nuc}} + \boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}] + \boldsymbol{\mathcal{F}}_{\text{mul}}\right)\end{aligned} \tag{3.78}$$

$$\begin{aligned}=& -\boldsymbol{\mu}_{\text{ind}}[\gamma^{e}]\left(\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\text{HF}}]}{\partial\xi} + \frac{\partial\boldsymbol{\mathcal{F}}_{\text{nuc}}}{\partial\xi}\right)\\ &-\boldsymbol{\mu}_{\text{ind}}[\gamma^{\text{HF}}]\frac{\partial\boldsymbol{\mathcal{F}}_{\text{el}}[\gamma^{\Delta}]}{\partial\xi}.\end{aligned} \tag{3.79}$$

The induction part of the correlated gradient can thus be evaluated similarly to the SCF case. It requires the solution of the induced dipole moments using the correlated density in the first term of eq (3.79), and the field derivative with the difference density in the second term. After all, these expressions are easily evaluated. Note that the above gradient expressions apply to all PTE-PE schemes, not only the PE-MP2 case. They also yield the correct PTE-PE-ADC(N) gradient by using the respective excited state relaxed densities.

3.4.4 PE Contributions to the LR-PE-ADC Gradient

When a LR coupling scheme is employed, the additional term in the PE-ADC Lagrangian is

$$\Delta\mathcal{L}^{\text{LR-PE-ADC}} = \sum_{klcd} x_{kl} x_{cd} V_{klcd}^{\text{ind,el}} = -\boldsymbol{\mathcal{F}}_{\text{el}}[\boldsymbol{\rho}^{(0)}]\mathbf{B}^{-1}\boldsymbol{\mathcal{F}}_{\text{el}}[\boldsymbol{\rho}^{(0)}]. \tag{3.80}$$

Carrying out the derivative is straightforward

$$\frac{\partial \Delta\mathcal{L}^{\text{LR-PE-ADC}}}{\partial \xi} = -2\boldsymbol{\mu}_{\text{ind,el}}[\boldsymbol{\rho}^{(0)}]\frac{\partial \boldsymbol{\mathcal{F}}_{\text{el}}[\boldsymbol{\rho}^{(0)}]}{\partial \xi}. \tag{3.81}$$

In the actual implementation, I use a symmetrized density matrix, yielding the following working equation

$$\frac{\partial \Delta\mathcal{L}^{\text{LR-PE-ADC}}}{\partial \xi} = -\frac{1}{2}\boldsymbol{\mu}_{\text{ind,el}}[\boldsymbol{\rho}^{(0)} + \boldsymbol{\rho}^{(0)\dagger}]\frac{\partial \boldsymbol{\mathcal{F}}_{\text{el}}[\boldsymbol{\rho}^{(0)} + \boldsymbol{\rho}^{(0)\dagger}]}{\partial \xi}. \tag{3.82}$$

To obtain the total PE contribution to the excited state LR-PE-ADC gradient, one needs to sum up the above expression and the expression for the PTE-PE-ADC gradient

$$\frac{\partial \Delta\mathcal{L}^{\text{LR-PE-ADC,tot.}}}{\partial \xi} = \frac{\partial \Delta\mathcal{L}^{\text{PTE-PE-ADC}}}{\partial \xi} + \frac{\partial \Delta\mathcal{L}^{\text{LR-PE-ADC}}}{\partial \xi}. \tag{3.83}$$

The PE contribution is then just added on top of the vacuum ADC gradient expression, yielding the correct PE-ADC energy derivative. Initial results demonstrating the correctness of my derivation and implementation can be found in Chapter 5. Note that the derivation I presented herein can be used to obtain gradients for related polarizable models, e.g., polarizable continuum models.

3.5 Summary of PE-ADC Methodologies

The derivations presented in this chapter for combining PE with ADC present a comprehensive toolkit to model all sorts of properties with the atomistic, polarizable model and an *ab initio* wave function method. Of note, all the equations are in principle transferable to other environment methods, which is particularly interesting for the derivation of analytic nuclear gradients with, e.g., the polarizable continuum model. At a glance, the derived methods and their possible applications are illustrated in the summary Figure 3.1, and the implementations are described in the following chapter.

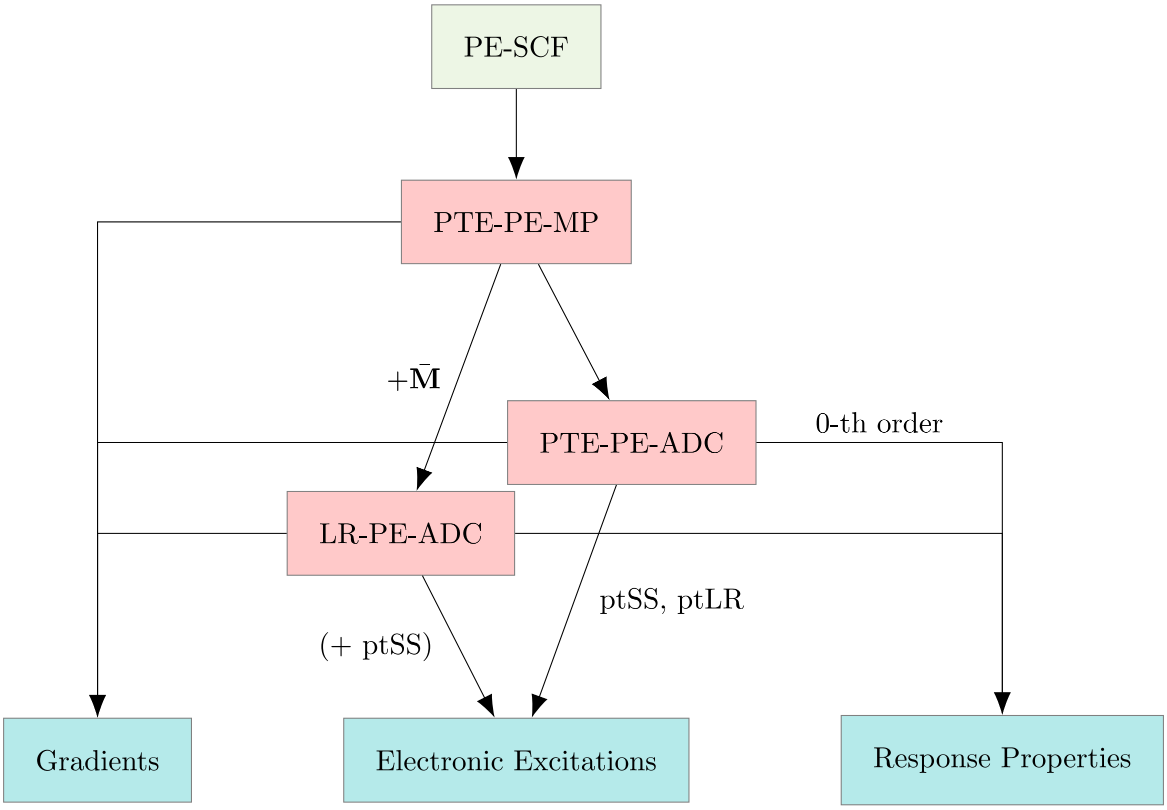


Figure 3.1: Illustration of PE-ADC methodologies (red) developed in this chapter and the resulting molecular properties that have become available (blue).

Chapter 4

Open-Source Libraries for Computational Spectroscopy

In the previous chapter, I outlined the theoretical foundations for the combination of the polarizable embedding model with ADC for the computation of various molecular properties. The development of such combined methodologies requires reliable, easy-to-use, and extensible software frameworks as completely orthogonal components, which need to be assembled and interfaced with one another. For example, the formulation of a PE framework is, in general, completely method-independent, and only the interface between PE and the specific electronic structure method of choice actually needs to "know" how quantities are to be handled. The realm of electronic structure theory and computational chemistry software has in recent years seen the emergence of more and more projects relying on hybrid approaches between high-level scripting languages like Python, and low-level compiled libraries written in C++.[38,81–83] During the work for this thesis, I have adopted the same approach to further the framework for computational spectroscopy using PE and ADC approaches, which resulted in mainly three software projects. First, I have implemented a standalone hybrid Python/C++ library for the polarizable embedding model, called CPPE. CPPE has enabled my own research work and it is currently interfaced to four different quantum chemistry programs, ready to be combined with quantum chemical methods of any kind.[79] Second, a toolkit for rapid development and experimentation of

---

Parts of this chapter have already been published in:

- M. Scheurer, P. Reinholdt, E. R. Kjellgren, J. M. H. Olsen, A. Dreuw, and J. Kongsted, "CPPE: An Open-Source C++ and Python Library for Polarizable Embedding", *J. Chem. Theory Comput.* **2019**, *15 (11)*, 6154-6163. (Reference 79)
- M. F. Herbst, M. Scheurer, T. Fransson, D. R. Rehn, and A. Dreuw, "adcc: A Versatile Toolkit for Rapid Development of Algebraic-Diagrammatic Construction Methods", *WIREs Comput Mol Sci.* **2020**, *10:e1462.* (Reference 80)

ADC-based methods was developed in collaboration with Dr. Michael F. Herbst in our group, which allowed for implementation of PE in combination with ADC.[80] Third, I have implemented the standalone `respondo` library that evaluates various response functions within the ADC/ISR framework as a part of the Gator project.[84] This library also contains state-of-the-art solver algorithms for efficient practical calculations of response properties. All three software projects, their individual design, and implementation will be described in the following.

4.1 CPPE: C++ and Python Library for Polarizable Embedding

A first step toward advanced environment modeling with *any* quantum chemical method is to have a reliable code for the environment part itself. The quantum and classical part are, to some extent, uncoupled from one another, i.e., their implementation can be achieved in an orthogonal manner. Hence, implementations of a *single* embedding method like the PE model can be modularized and interfaced to multiple quantum chemical program packages, and be combined with a multitude of methods therein.[79,85,86] Exploiting modular libraries puts the main focus back on the development of quantum mechanical methods: Using a well-tested, production-ready library for the inclusion of environment contributions is much more sustainable and time-efficient than re-implementations for each program package and method. The modular library must be cautiously designed to avoid inefficient and overly complicated interfaces in the future. For this reason, it is important to know exactly which tasks are to be performed by the library and which not. One may now argue that such a library for the PE model already exists: PElib[87] is a Fortran library with which the original implementation of the PE model formulation used also in this thesis was achieved.[14,15] PElib is certainly the most used implementation currently publicly available, its source is fully disclosed, and it is interfaced to the Dalton program,[88] where it enables PE computations with an enormous amount of methods.[27,28,30,89–92] Due to close “entanglement” with the host program Dalton, PElib is not necessarily the most suitable solution to provide modularity, extensibility, and maintainability. Furthermore, no ADC-based methods are available in the programs that work with PElib. Motivated by these problems and inspired by the feature-rich PElib, an initial standalone PE implementation in Q-Chem,[93,94] fully written in C++, was achieved, enabling my work in Ref. 61. Going a step further, this pilot implementation[42] was refactored and led to a fully open-source modular library for PE, named CPPE, that is the “C++ and Python library for PE".[79] The CPPE library contains the necessary routines to implement ground state and molecular property calculations with PE, previously outlined in Chapters 2 and 3. Through its minimalist application programming interface (API), CPPE can be easily coupled to any quantum mechanical host program with minor programming effort. The library was inspired by PCMSolver[86] and libefp,[85] which are similar libraries for continuum solvation models and the effective fragment potential (EFP) method, respectively, and of course, PElib. Since its first release, I have interfaced CPPE to four quantum chemical program packages: Q-Chem,[93,94] Psi4,[38,81] PySCF,[39,95] and VeloxChem.[83] All implemented interfaces led to co-authorship on the respective publications of the program packages.[38,39,83,94]

CPPE provides both a C++ and a Python API, exposing the necessary high-level functionality. The Python API allows for quick manipulation of data and rapid prototyping to try out new variants or combinations of the PE model, and is used in all interfaces except for Q-Chem. On the low level, CPPE was originally based on PElib[87] and still contains some code that was "literally" translated from Fortran to C++, however, I have refactored most of the routines by now to adopt a more C++-typical code style. CPPE is designed to be as modular as possible, such that it can be interfaced to *any* host program. Therefore, the CPPE code base is completely free of any host-program-specific code. CPPE has already rendered calculations with PE more accessible as it becomes available in more and more program packages. In the following, I explain the design philosophy of CPPE, together with implementation details considering the theoretical methodology from Section 2.5.1. After that, I give instructions on how to easily interface CPPE to a new quantum chemistry program, together with specific examples of the host programs I have interfaced it to in the past.

4.1.1 Design and Implementation

Implementation of the CPPE library was initially guided by the existing Fortran library PElib.[87] PElib was also used as a reference for quick testing of the individual routines in the course of the implementation. I aimed for high modularity, host program agnosticy,[86] and extensibility to achieve a sustainable library design. Therefore, CPPE is implemented in C++, which provides the necessary scaffold for data containers and standard algorithms through the standard template library (STL) and object-oriented programming. The latter makes the implementation of the necessary data structures for the PE library intuitive and easily extensible. CPPE is built with CMake,[96] widely used in quantum chemical program packages, e.g., in Psi4,[38,81] and thus making CPPE easy to integrate in an existing build setups as an optional external dependency. The CMake setup in CPPE was adopted from other external libraries that are optional dependencies of Psi4. To perform numerical linear algebra operations, e.g., solving the linear equations for induced moments, the header-only Eigen3 library[97] is used. To bring CPPE to modern Python-based quantum chemistry codes, the functionality of the library is interfaced to the Python layer by the lightweight header-only pybind11 library.[98] The pybind11 interface code inside CPPE is very concise and allows for rapid extension of Python-exposed functionality. Other than the code of the testing framework and some scripts, CPPE does not contain any Python code needed for functionality. Consequently, new functions and features are implemented in C++ and then exposed to Python via an appropriate interface. Furthermore, Eigen3 matrices and vectors are seamlessly converted to `numpy` arrays[99] and vice versa through pybind11. A set of unit and functionality tests are implemented in Python using `pytest`.[100] In general, the presented hybrid C++/Python programming approach has also proven powerful in the recently published Psi4NumPy package.[82] A summary of the number of source code lines per programming language for the current CPPE version is shown in Figure 4.1. As can be seen in the bar plot, CPPE relies on autogenerated code for some parts of the library, contributing most of the source code. Other than that, the library is rather lightweight with less than 700 lines of Python code for testing and less than 3000 lines of handwritten C++ code for the core functionality and pybind11 interfaces. For this reason,

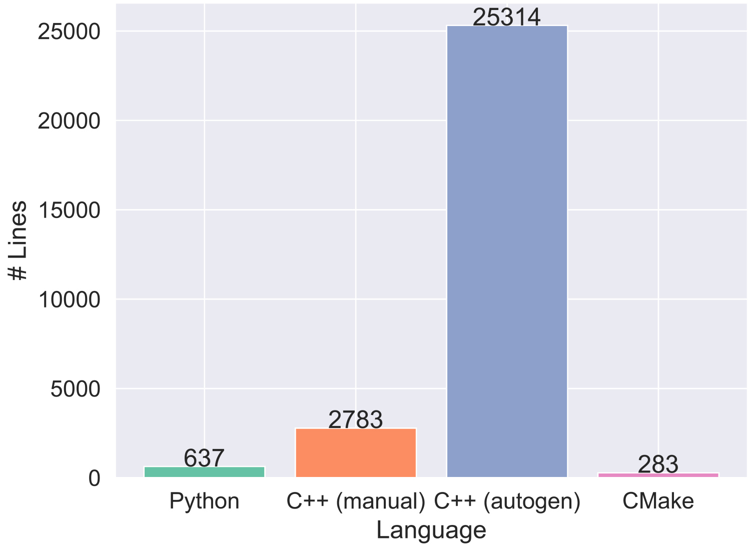
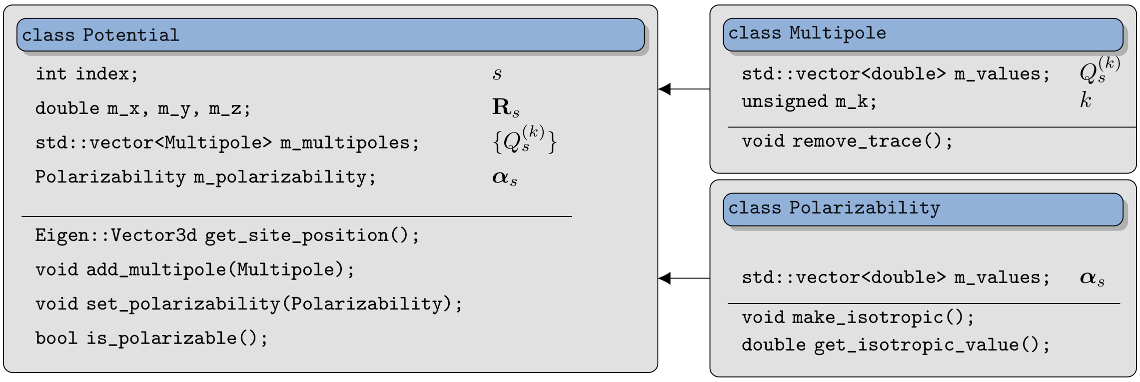


Figure 4.1: Number of source code lines for each language in CPPE, version 0.3.1. The C++ statistics include both header and source files.

it is easy for new developers to get started and to identify how the library components are connected to each other. The most recent version of the CPPE source code (`v.0.3.1`) can be downloaded from GitHub (`https://github.com/maxscheurer/cppe`). The C++ core library is contained in the `cppe/core` directory in the downloaded folder, whereas the Python bindings reside in `cppe/python_iface`. Since recently, CPPE is deployed to the Python Package Index for easy installation via `pip install cppe`, and installation via the Anaconda package manager with the command `conda install -c conda-forge cppe` is conveniently possible. The deployment of a new CPPE version is automatically run once a new release is tagged on GitHub, reducing the work for the maintainer dramatically.

Most importantly, the core library is equipped with data containers for embedding potentials, comprised of multipole moments ($\{Q_s^{(k)}\}$, `Multipole` class) and polarizabilities ($\{\boldsymbol{\alpha_s}\}$, `Polarizability` class). The main parameter container for each site in the environment is the `Potential` class, comprised of coordinates $\mathbf{R}_s$ for a site s, together with a list of multipoles (`std::vector<Multipole>`), the polarizability (`Polarizability`), and some helper functions. The full environment parametrization is stored as a `std::vector<Potential>`. The three parameter container classes are depicted in Figure 4.2. Usually, all fields and functions of such classes are exposed to Python in a “pythonic” manner, i.e., the code reads nicely next to real Python code and does not make the impression to be C++ under the hood. CPPE reads the parameters mentioned above from



```
class Potential
int index;                                  
double m_x, m_y, m_z;                       
std::vector<Multipole> m_multipoles;        
Polarizability m_polarizability;            

Eigen::Vector3d get_site_position();
void add_multipole(Multipole);
void set_polarizability(Polarizability);
bool is_polarizable();
```

```
class Multipole
std::vector<double> m_values;
unsigned m_k;

void remove_trace();
```

```
class Polarizability
std::vector<double> m_values;

void make_isotropic();
double get_isotropic_value();
```

Figure 4.2: C++ classes containing the embedding potential. The `Multipole` and `Polarizability` classes contain the actual parameters and provide helper functions, e.g., removing the trace of a multipole moment or making a polarizability isotropic. A single site in the environment is fully parametrized through an instance of the `Potential` class, consisting of the site index s, the coordinates, and vectors of multipoles and polarizabilities. Helper functions make it easy to add additional parameters or to check if a specific site is polarizable.

a so-called potential file. The format is identical to that used in PElib, explained in Ref. 18. Read-in is performed by the `PotfileReader` class. If a special treatment of the border between the quantum and classical region is required, the `PotManipulator` class can, for example, redistribute or remove parameters of the affected sites. In addition to parameters, information about the quantum region has to be stored, for example, to evaluate the nuclear electrostatic interaction energy or gradient (eqs (2.132) and (2.157)). This is achieved by the `Atom` class, which holds the coordinate $\mathbf{R}_n$, charge Z_n, and element symbol of an individual atom in the quantum region. A `Molecule` class is then defined as a type alias of `std::vector<Atom>`. The core library naturally provides classes to compute all classical energies and electric fields: `MultipoleExpansion` evaluates the nuclear-multipole interaction energy (eq (2.132)), whereas `NuclearFields` and `MultipoleFields` implement electric field contributions from nuclei and multipole moments, respectively (eqs (2.139) and (2.140)). These classes also handle the computation of classical contributions to the analytical nuclear gradient.

For solving the system of linear equations of the induced dipole moments in eq (2.143), an iterative conjugate gradient (CG) solver is employed. In the solver procedure, explicit construction of the $\mathbf{B}$ matrix is not needed, but the algorithm is implemented in a contraction-based manner, i.e., only using the matrix-vector product with a trial vector. This is taken care of by the `BMatrix` class, which contains routines to apply the $\mathbf{B}$ matrix to a vector (`apply` function), or the inverse diagonal for Jacobi preconditioning (`apply_diagonal_inverse` function). Explicit construction would be computationally prohibitive for large environments, however, the `BMatrix` class implements this feature for testing purposes. The solver algorithm itself is implemented in the `InducedMoments` class, providing convenient wrappers to solve equations like (2.143) with any right-hand side. A guess vector for the linear equations is obtained by applying the preconditioner once to the right-hand side, i.e., multiplying the field vectors at all sites with the respective polarizability. As can be seen from the theoretical part, the most important building block of the classical expressions are the T-tensors (eq (2.133)), which are usually computed using an open-ended formula.[18,101] For low-order T-tensors, however, it is beneficial for performance to generate the code for evaluation of T-tensors automatically. This was done based on a Python script written by my collaborator P. Reinholdt as part of his `polarizationsolver` project (`https://gitlab.com/reinholdt/polarizationsolver`). The script to generate the code for T-tensors up to a certain order can be found in the `scripts/gen_tensors` directory of the CPPE source. The script evaluates the derivatives of $1/R$ analytically with SymPy[102] producing the necessary source and header files for direct integration in CPPE. The function to evaluate the analytic derivative is shown

```python
def T_cart(sequence):
    x, y, z = sympy.symbols("x y z")
    R = sympy.sqrt(x**2 + y**2 + z**2)
    T = 1 / R
    if sequence:
        return sympy.diff(T, *[i for i in sequence])
    else:
        return T
```

Listing 4.1: Code fragment to produce Cartesian T-tensors with SymPy by analytic differentiation.

in Listing 4.1. For optimized performance, common subexpression elimination (CSE) is enabled, and the power function `std::pow(b, n)` is expanded to multiplications up to $n = 5$. In the current version of the code, T-tensors are generated up to sixth order as normally no higher-order tensors are required in practice. If the user desires to compute tensors of higher order, the recursive, open-ended code can be used for that. To avoid over-polarization, interactions involving T-tensors, i.e., permanent multipole fields (eq (2.140)), or dipole-dipole interaction tensors ($\mathbf{T}^{(2)}$ in eq (2.144)) can be damped using Thole's exponential scheme.[103,104] The code for the damped tensors is autogenerated, too.

User-provided options, e.g., the path of the potential file, convergence thresholds, or treatment of the border between the quantum and classical region, are defined in the `PeOptions` class. On the Python side, it is, however, much more natural to handle options and parameters using dictionaries. For this reason, the methods that require a `PeOptions` instance as argument on the C++ level are wrapped, such that a Python `dict` can be conveniently passed. Since Python dictionaries are fully dynamic, the keyword names and their respective types are checked for consistency in the wrapper function. The inverse wrapper also exists, such that the options passed to CPPE can be conveniently printed as a `dict` in Python code.

The aforementioned low-level building blocks and functions do not need to be assembled from scratch when interfacing CPPE to a new host program, which would be tedious and error-prone: CPPE provides a convenient top-level wrapper of all low-level functions exposed through the `CppeState` class (Fig. 4.3). Using the `CppeState` to manage all necessary PE tasks reduces the programming effort because all implemented functions and data fields of `CppeState` are self-explanatory and correspond to the equations given

`class CppeState`	
`Molecule m_mol;`	quantum region molecule
`std::vector<Potential> m_potentials;`	potentials for all sites
`Eigen::VectorXd m_nuc_fields;`	$\mathcal{F}_{\text{nuc}}$
`Eigen::VectorXd m_multipole_fields;`	$\mathcal{F}_{\text{mul}}$
`Eigen::VectorXd m_induced_moments;`	$\boldsymbol{\mu}_{\text{ind}}$
`PeOptions m_options;`	option container
`CppeState(PeOptions options, Molecule mol);`	explicit constructor
`void calculate_static_energies_and_fields();`	calculates $\mathcal{F}_{\text{nuc}}$, $\mathcal{F}_{\text{mul}}$, and $E^{\text{nuc}}_{\text{es}}$
`void update_induced_moments(...);`	solves $\mathbf{B}\boldsymbol{\mu}_{\text{ind}} = \mathcal{F}$
`double get_total_energy();`	returns E_{PE}
`std::string get_energy_summary_string();`	returns a string with all energy contributions
`Eigen::MatrixXd nuclear_interaction_energy_gradient();`	returns $\mathbf{E}^{\text{nuc},a}_{\text{es}}$
`Eigen::MatrixXd nuclear_field_gradient();`	returns $\mathcal{F}^{a}_{\text{nuc}}$

Figure 4.3: `CppeState` class members. The `CppeState` serves as the top-level interface of the CPPE library. It exposes a variety of functions to carry out all host-program-independent tasks. All the building blocks, e.g., to solve the linear equations for induced dipole moments, are properly assembled in the `CppeState` functions. Further, `CppeState` manages bookkeeping of energy contributions and electric fields.

in the Theory part. The `CppeState` can be constructed from a `Molecule` object and a `PeOptions` object. After instantiation, the potential file is automatically parsed, manipulated if needed, and stored inside the `CppeState`. Then, the `CppeState` instance is ready for use until it is deallocated. Through such a high-level class that allows access to all required routines, some flexibility is of course given away, but the individual components of CPPE can still be used in a dynamic workflow. With the components and high-level interface of CPPE API at hand, I now want to provide a small tutorial on how to implement the host program side of the interface in a general manner.

4.1.2 How to Interface CPPE to a Host Program

CPPE is completely agnostic of any host-program-specific code and data. As a result, only an interface on the host program side needs to be implemented, integrating both CPPE and program-specific routines. A schematic overview of the overall interface structure is shown in Figure 4.4. The host program side of the interface communicates with the input reader, integral library, SCF driver, and post-SCF drivers of the host program. A mock implementation of such a `CppeHostProgramInterface` is presented using Python code snippets. First of all, the constructor of the class (Listing 4.2) takes a `Molecule` and `PeOptions` object to build the initial `CppeState`. Furthermore, the function constructs a `numpy` array with the coordinates of all polarizable sites for downstream computation of field integrals. Finally, the static contributions to the electrostatic interaction energy and electric fields are calculated. The key ingredient of the host program interface is to expose a routine to compute the PE operator and energy from an input density matrix. Such a density-driven function can be employed both in the SCF driver and a post-SCF driver, as previously explained in Chapter 3. For gradients, it is more convenient to implement an additional function, handling exclusively gradient contributions, into the host program interface. An illustrative implementation of the PE contribution routine is displayed in Listing 4.3. The PE contribution routine first needs to compute the electrostatics operator (eq (2.135), step I), making use of the host program integral library together with the multipole moments stored in the `CppeState`. A sketch of this function inside the `CppeHostProgramInterface` class is displayed in Listing 4.4.

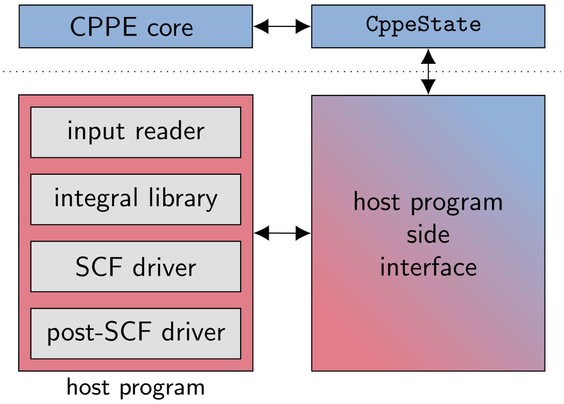


Figure 4.4: Schematic overview of the interface structure. The host program side of the interface to CPPE requires access to the input reader (if applicable), integral library, SCF driver, and post-SCF driver of the host program. The host-program-independent tasks are taken care of on the CPPE side, wrapped by a state object. Since the interface to include the CPPE state is minimal, the major programming work is in gathering the required data and integrals from the host program.[1]

The required integrals (eqs (2.136) and (2.147)) must be available in the host program. For

```python
import numpy as np
from integral_library import Integrals  # mock host program integral library
from cppe import CppeState

class CppeHostProgramInterface:
    def __init__(self, molecule, options):
        self.cppe_state = CppeState(molecule, options)
        # coordinates of polarizable sites
        self.polarizable_coords = np.array([
            site.position for site in self.cppe_state.potentials
            if site.is_polarizable
        ])
        # initialize the host program integral library
        self.integrals = Integrals()
        self.cppe_state.calculate_static_energies_and_fields()
```

Listing 4.2: Mock constructor of `CppeHostProgramInterface`.

practical applications, however, it is often sufficient to have potential derivative integrals through second order, i.e., normal Coulomb integrals (used for the nuclear attraction operator), electric field integrals, and electric field gradient integrals. With these features, it is possible to model electrostatic interactions up to quadrupole moments and to employ self-consistent treatment of dipole polarization. Once computed, the electrostatics operator can be cached. After storing the operator, the electronic contribution to the electrostatic interaction energy (eq (2.134)) is obtained as the product-trace with the density matrix (line 5 in Listing 4.3). Second, the PE routine obtains the expectation values of the electric field operator from the input density matrix (step II) to compute the total electric field at all polarizable sites (eq (2.138)). Third, a simple call to the `CppeState` is made, requesting induced dipole moments from the given electric fields (step III). In the background, the system of linear equations (eq (2.143)) is solved, the induction energies are updated (eq (2.137)), and the resulting induced dipole moments are returned as a `numpy` array. In the fourth step, the induction operator (eq (2.145)) is formed by contracting the electric field integrals with the induced dipole moments (step IV). If the flag `elec_only` is set to `True`, only electronic contributions to the energy and PE operator are taken into account, as it is required for post-SCF procedures (e.g., eq (3.13)). Otherwise, the full operator (eq (2.150)) is assembled and returned, together with the PE energy contribution (lines 21-27 in Listing 4.3). Due to the simple structure of the host program interface, only a single routine needs to be called from all places in the program where PE contributions are required, as illustrated in Figure 4.5. For example, the SCF driver of the host program

```python
def get_pe_contribution(self, density_matrix, elec_only=False):
    # step I: build electrostatics operator
    if not self.V_es and not elec_only:
        self.build_electrostatics_operator()
    e_electrostatic = np.sum(density_matrix * self.V_es)
    self.cppe_state.energies["Electrostatic"]["Electronic"] = e_electrostatic

    # step II: obtain expectation values of elec. field at polarizable sites
    elec_fields = self.integrals.electric_field_value(
        self.polarizable_coords, density_matrix
    )
    # step III: solve induced moments
    self.cppe_state.update_induced_moments(elec_fields)
    induced_moments = self.cppe_state.induced_moments

    # step IV: build induction operator
    V_ind = np.zeros_like(self.V_es)
    for coord, ind_mom in zip(self.polarizable_coords, induced_moments):
        field_int = self.integrals.electric_field_integral(site=coord)
        V_ind += -1.0 * sum(ind_mom[i] * field_int[i] for i in range(3))
    E_pe = self.cppe_state.energies.total_energy
    V_pe = self.V_es + V_ind
    # only take electronic contributions into account
    if elec_only:
        V_pe = V_ind
        E_pe = self.cppe_state.energies["Polarization"]["Electronic"]
    return E_pe, V_pe
```

Listing 4.3: Mock PE contribution routine of the `CppeHostProgramInterface` class

```python
def build_electrostatics_operator(self):
    n_bas = self.integrals.n_bas  # number of basis functions
    self.V_es = np.zeros((n_bas, n_bas))  # zero numpy array for operator matrix
    for site in self.cppe_state.potentials:
        for multipole in site.multipoles:
            self.V_es += self.integrals.potential_derivative(
                position=site.position, order=multipole.k,
                moments=multipole.values
            )
```

Listing 4.4: Mock code snippet for computation of V_{pq}^{es}.

calls `get_pe_contribution` every iteration providing the current SCF density matrix, and in turn receives the PE contribution to the Fock operator without further ado. This sim-

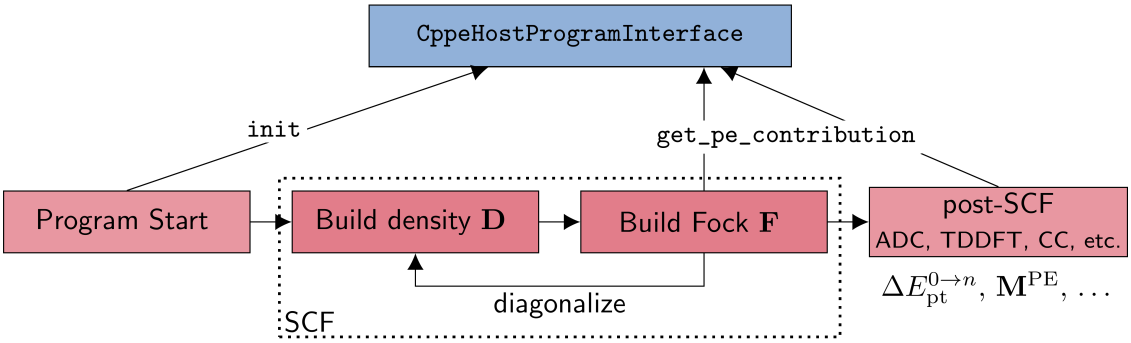


Figure 4.5: Overview of a program workflow employing PE via the `CppeHostProgramInterface`. After the host program (red boxes) starts, the interface is initialized. In downstream routines, e.g., the SCF procedure or post-SCF methods, only the `get_pe_contribution` function needs to be called providing any kind of input density matrix.[1]

plistic and clean design makes it easy to implement PE contributions in various places in the host program with a single function call. Furthermore, it makes the interface code easily maintainable. Note that the here presented interface is over-simplified for illustration purposes, some more details of the actually implemented host program interfaces are given below. The CPPE library is currently interfaced to four program packages – Psi4, PySCF, Q-Chem, and VeloxChem. All four host program interfaces are rather concise and could be implemented with minimal time effort according to the how-to guide presented above. Particularities and important aspects of the individual interfaces are discussed in the following.

4.1.3 Interface to Psi4

The interface from Psi4 to CPPE is fully implemented in Python. The advanced CMake setup of Psi4 makes it possible to integrate source builds of CPPE directly, or to use a separately built CPPE, i.e., the library must only be present at run time. From a technical point of view, the integration into Psi4 was rather easy because an interface for the related EFP method with libefp[85] already existed. Hence, the multipole potential integrals were readily accessible. Along the way, I had to refactor some parts of the Psi4-internal integral library to enable a) evaluation of multipole potential integrals only through a certain order (`https://github.com/psi4/psi4/pull/1657`), b) speed-up of electric field computations (`https://github.com/psi4/psi4/pull/1671`), and c) integral routines for a pilot PE gradient implementation (`https://github.com/psi4/psi4/pull/2039`). This is one of the great advantages of open-source codes, because in the end, both users and all developers benefit from the joint effort. Using the Psi4 and Psi4NumPy API, it is rather easy to amend user code with simple PE-related computational tasks. I want to illustrate this here with the example of perturbative state-specific corrections of electronic excitation energies, which I derived in Section 3.1. Regarding electronic structure methods for excited states, Psi4 provides a selection of equation-of-motion coupled cluster (EOM-CC). Densities of the excited states are readily available. With the CPPE interface to Psi4 in place, it is rather simple to implement ptSS corrections for EOM-CC excited states using Psi4NumPy. The corresponding Python code snippet is shown in Listing 4.5. A converged coupled cluster wave function object `ccwfn`, together with the number of excited states `nroots` is passed to the Python function. Inside the loop over states, an energy correction is computed for each individual excited state based on the CC difference density matrix, as shown in eq (3.6). The here presented code was used in Ref. 79 to model the excited states of a chromophore embedded in a protein environment with PE-EOM-CC2. In the same manner, it was possible to add LR-PE contributions[66] to time-dependent density functional theory (TDDFT) or linear response computations to Psi4 (`https://github.com/psi4/psi4/pull/2006`). Since the overall structure of PE and PCM is rather similar and the working equations do not differ except for which ‘kernel’ is used to evaluate the contraction with the density matrix, adding LR-PE and LR-PCM capabilities was possible in one go. Hence, in the process of maintaining the CPPE interface inside Psi4, I have also seized the opportunity to improve the code in regard to other solvent models, refactored

```python
def compute_ptss_corrections(ccwfn, nroots):
    ptss = []
    for i in range(1, nroots + 1):
        # obtain the CC density matrix of state i
        ccdmat = ccwfn.variable(f"CC ROOT {i} DA")
        # obtain the SCF density matrix
        scfdmat = ccwfn.Da()
        # compute the difference density
        ccdmat.subtract(scfdmat)
        # scale with 2 due to restricted reference
        ccdmat.scale(2.0)
        # compute the energy correction
        energy, _ = ccwfn.pe_state.get_pe_contribution(ccdmat.np, elec_only=True)
        ptss.append(energy)
    return ptss
```

Listing 4.5: Computation of ptSS corrections for EOM-CC states with Psi4NumPy and CPPE.

and cleaned up parts of the EFP integration code, and enabled new features not directly related to PE. To avoid electron spill-out in PE computations,[105] I have also added the recently presented PE(ECP) repulsive potentials to the Psi4 host program interface.[106]

4.1.4 Interface to PySCF

The CPPE interface to PySCF is entirely written in Python. No integration with the build system is actually needed due to the various ways of installing CPPE, i.e., it is also a run time dependency of PySCF. The host program interface resides in a single source file inside PySCF, namely `pyscf/solvent/pol_embed.py`. A particularity of the PySCF implementation is the evaluation of the integrals: the underlying integral library does not have the functionality to carry out higher order derivative on the 'mid' $1/R$ part of the integral (e.g., eq (2.136)), such that partial integration must be used in order to shift the derivative from the $1/R$ part to the bra and ket side. Consider the following general integral

$$I_k = \int \phi_p^*(\mathbf{r}) \nabla^k \frac{1}{|\mathbf{r} - \mathbf{R}|} \phi_q(\mathbf{r}) \mathrm{d}\mathbf{r} \tag{4.1}$$

with some arbitrary k-th order derivative of the Coulomb potential. For practical calculations with PE, usually only potentials through quadrupoles, i.e., $k = 2$, are required. Performing integration by parts in the above equation gives

$$\begin{aligned} I_k = &\underbrace{\nabla^{k-1} \frac{1}{|\mathbf{r} - \mathbf{R}|} \phi_p^*(\mathbf{r}) \phi_q(\mathbf{r}) \Big|_{-\infty}^{\infty}}_{=0} \\ &- \int \nabla \phi_p^*(\mathbf{r}) \nabla^{k-1} \frac{1}{|\mathbf{r} - \mathbf{R}|} \phi_q(\mathbf{r}) \mathrm{d}\mathbf{r} \\ &- \int \phi_p^*(\mathbf{r}) \nabla^{k-1} \frac{1}{|\mathbf{r} - \mathbf{R}|} \nabla \phi_q(\mathbf{r}) \mathrm{d}\mathbf{r}. \end{aligned} \tag{4.2}$$

Another partial integration gives

$$
\begin{aligned}
I_k = & \underbrace{- \nabla^{k-2} \frac{1}{|\mathbf{r}-\mathbf{R}|} \nabla \left(\phi_p^*(\mathbf{r}) \phi_q(\mathbf{r}) \right) \Big|_{-\infty}^{\infty}}_{=\,0} \\
& + \int \nabla^2 \phi_p^*(\mathbf{r}) \nabla^{k-2} \frac{1}{|\mathbf{r}-\mathbf{R}|} \phi_q(\mathbf{r}) \mathrm{d}\mathbf{r} \\
& + \int \phi_p^*(\mathbf{r}) \nabla^{k-2} \frac{1}{|\mathbf{r}-\mathbf{R}|} \nabla^2 \phi_q(\mathbf{r}) \mathrm{d}\mathbf{r} \\
& + 2 \int \nabla \phi_p^*(\mathbf{r}) \nabla^{k-2} \frac{1}{|\mathbf{r}-\mathbf{R}|} \nabla \phi_q(\mathbf{r}) \mathrm{d}\mathbf{r}.
\end{aligned}
\tag{4.3}
$$

Thus, one can shift the original derivative to the derivatives on the basis functions, which are available in PySCF. Since recently, the implementation in PySCF can be used for rather large environments: to achieve this, the computation of integrals is batched according to the available memory, such that bottlenecks through too large integral matrices when the number of environment sites is large are avoided (`https://github.com/pyscf/pyscf/pull/700`). The PE implementation in PySCF is, due to efficient and threaded integrals, probably the fastest one using CPPE. It also contains the PE(ECP) feature mentioned above.[106]

4.1.5 Interface to Q-Chem

The interface to Q-Chem was the first one established, and it is written purely in C++. Ground state PE-SCF calculations are available together with the simulation of excited states using the pt-PE-ADC method.[61] In principle, it is possible to combine PE with any post-SCF method in Q-Chem, e.g., EOM-CC methods while using PTE. For the EFP method, which is the predominantly used and promoted explicit solvent model of Q-Chem, this procedure is common and has been employed for example for TPA calculations.[70] The same calculations in Q-Chem are also possible with the PE model. The first application project of CPPE was to model excited states with pt-PE-ADC for large biomolecular environments.[61]

4.2 ADC-connect: Toolkit for Rapid Development of ADC Methods

With a PE implementation in open-source Python-based quantum chemical libraries available, there is still an important component missing for computational spectroscopy, namely the electronic structure method of choice: The ADC family of methods has only gained little attention in recent endeavors to create open-source libraries for such purposes. In Psi4 and PySCF, for example, focus is put much more on coupled cluster methodologies. Recently, ADC methods for computation of ionization and electron attachment energies were added to PySCF.[107] A comprehensive toolkit to quickly carry out the implementation of combined PE-ADC methodologies described in the previous Chapter 3 was however still missing. Of course, such a toolkit's purpose is not to implement ADC in combination with environment methods only. It should enable rapidly prototyping of new ADC variants, and be competitive with established codes for production calculations. To this end, our group developed the ADC-connect toolkit, abbreviated `adcc`, as a versatile toolkit for the rapid development of ADC methods.[80] As for the CPPE library and Psi4, we pursued the same hybrid Python/C++ programming strategy to allow for high-level workflow and method design in Python user code as well as performant tensor operations in C++ "under the hood" using the `libtensor` library.[108] All the data structures are conveniently available on the Python layer and can be used with, e.g., NumPy or `matplotlib`.[99,109] In late 2020, the `adcc` toolkit became a fully open-source package that can be freely downloaded and used. The `adcc` library is fully standalone and can be seamlessly connected to quantum chemical host programs to run ADC calculations on top of the provided SCF reference state. Existing software packages of the Python ecosystem are employed for common tasks, e.g., linear algebra operations, data export, visualization, and other post-processing tasks. By re-using these libraries, the development of `adcc` focuses completely on the implementation of ADC methods, and all other related tasks are handled by third-party libraries, providing full flexibility. This is a stark contrast to classical monolithic quantum chemistry software, which usually consists of a single code base. The interface design is at the heart of the entire `adcc` toolkit, and has, in my opinion, proven easy to use and maintain over the past years. Currently, interfaces to the host programs Psi4,[38] PySCF,[39] `molsturm`,[110] and VeloxChem[83] are implemented. In this section, I present the design strategy of `adcc` as published in Ref. 80, the structure of the library, my code and feature contributions to `adcc`, including the main changes since the first publication.

4.2.1 Design Goals

The aim to implement a standalone, flexible library for ADC development requires a specific design for seamless integration into the Python ecosystem, while carefully retaining the specific needs and features for wave function based electronic structure calculations. These design aspects, presented already in the publication,[80] are outlined in the following. First, a common strategy is to rely on established software packages of the existing ecosystem rather than building a full-featured quantum chemistry package from scratch. The `adcc` toolkit should only fulfill the unique purpose of providing functionality for ADC itself, and nothing else. Re-using the existing quantum chemical programs available in Python, i.e., years of development work of the large Psi4 and PySCF community, is ecological and keeps `adcc` a light-weight code, orthogonal to other endeavors. The second design goal to provide an open and simple interface for reproducibility is a direct consequence: `adcc` must provide a simple and clean interface for any SCF host program on the Python layer to enable interoperability among several codes. The user can interchange host programs as desired to verify results across different SCF implementations. The key workflows implemented with `adcc` are automatically agnostic of the host program and can be used in a plug-and-play fashion with any supported code, a design principle I have explained in detail for CPPE. The third design goal to find a good compromise between code complexity and performance balances the needs of developers and end users of the library. It should be possible to implement new features with ease while still being able to run productive ADC calculations for practical purposes. Usually, performant electronic structure codes come with high code complexity and reduced code readability. This problem is solved to some extent automatically through the hybrid Python/C++ approach. All workflow components that are needed to steer, modify, or extend `adcc` are found on the Python layer, including the entire working equations required for ADC. On the other hand, routines that are performance bottlenecks, i.e., import of SCF data like electron-repulsion integrals (ERIs) and highly demanding tensor operations are handled by parallelized C++ code through an abstract tensor interface to `libtensor`. Note that this general tensor interface could in principle be used with other tensor libraries as well. As a consequence, the fourth and last main goal of `adcc`'s design is to maintain a low barrier between users and developers. Most of the users might start out just wanting to run ADC calculations for their research. They might, however, stumble upon a missing feature or want to integrate `adcc` in a more complex workflow of their own. Through the open and detailed documentation (`https://adc-connect.org`) we want to enable easy understanding such that simple programming tasks are possible also for new developers.

Through community-driven development, `adcc` would be possible to grow beyond what a single developer or scientist can imagine. Making `adcc` a fully open-source was the final step to an entirely open platform for ADC method development, residing completely on GitHub and using freely available developer tool chains such as continuous integration (CI) and continuous deployment (CD).

4.2.2 Demand-Driven Workflow and Library Structure

An illustration of the common workflow of an ADC calculation inside `adcc` is depicted in Figure 4.6. As explained before, `adcc` is *not* responsible for carrying out the ground state SCF calculation and the required data. Instead, this is handled by a (third-party) SCF code, called host program herein. The terminology is identical to the one used in the description of the CPPE library. In Figure 4.6, the `adcc`-specific classes and functions are shown in `teletype` font. When providing a converged SCF state to the `run_adc` function, or pre-defined aliases like `adc2`, the library selects an appropriate `HartreeFockProvider` implementation to gain access to all required reference state data. The `HartreeFockProvider` defines a unique and clean interface such that no host-program-specific routines are spilled over into `adcc`. All data that need to be imported to run ADC calculations, e.g., ERIs, MO energies, MO coefficients, operator integrals, and many more, are handled via the custom subclass of `HartreeFockProvider`.[80] Details on how to implement a program-specific `HartreeFockProvider` can be found in the `adcc` publication[80] or in the `adcc` online documentation (`https://adc-connect.org`). Another important component is the container for the tensor block splitting, `MoSpaces`, capable of mapping host-program-specific indices to `adcc`-internal MO blocks (`o` for occupied, `v` for virtual in the following). The `MoSpaces` object is also required to set up custom tensors

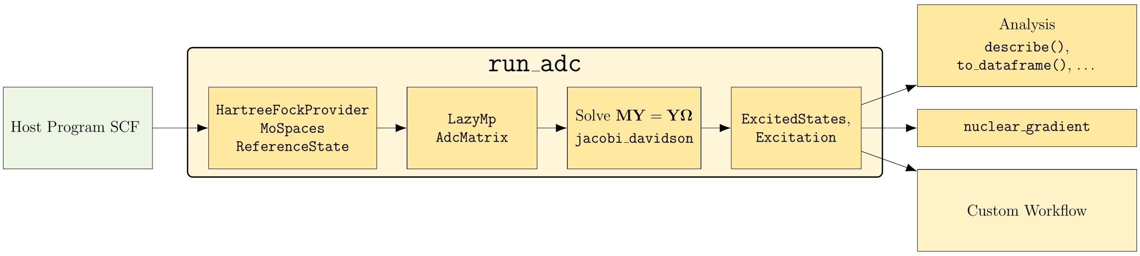


Figure 4.6: Schematic illustration of the standard `adcc` workflow. The `run_adc` routine takes an SCF result from a host program, prepares the solution of the eigenvalue problem and yields the `ExcitedStates` or `Excitation` object as result containers. These can be used for further built-in analysis, nuclear gradients, or customized workflows in user code.

and operators. Finally, the first block consists of constructing the `ReferenceState` class that defines the top-level `adcc`-side data container for all SCF-related data, ubiquitously used in the entire toolkit. For example, let `hf` be an instance of `ReferenceState` imported from a specific host program. Then `hf.energy_scf` returns the SCF energy of the molecule, `hf.foo` is a short-cut to get the `oo` block of the Fock matrix, and `hf.oovv` retrieves the $\langle \mathsf{oo} || \mathsf{vv} \rangle$ integral. The possibility to conveniently include environment models is granted by `ReferenceState`, as will be explained later. As a next step, the bookkeeping classes for the MP ground state, `LazyMp`, and the lazy representation of the ADC matrix, `AdcMatrix`, are created.[80,110,111] The `AdcMatrix` class collects all previous instances of data containers (i.e., reference state and MP results) and in addition wraps the working equations of the matrix-vector product and matrix diagonal for a specific ADC(N) scheme. Both the `ReferenceState` and `LazyMp` objects are of course used throughout the working equations. Once the eigenvalue problem (eq (2.51)) is solved through the iterative solver, all excited state information is conveniently wrapped in an `ExcitedStates` object. In addition, properties of *individual* excitations are available via the `Excitation` object. These attributes of `Excitation` are automatically generated from the parent `ExcitedStates` class by Python decorators in order to avoid code duplication. Convenient member functions of the `ExcitedStates` instance are available for built-in analysis. A summary table with energies, oscillator strengths, and dipole moments can for example be printed with the `describe` function. Furthermore, translation of the `ExcitedStates` object to a `pandas.DataFrame` is implemented.[112,113] This enables custom data visualization and manipulation in user code. Analytic nuclear gradients are currently available in a development version of `adcc` (see `https://github.com/adc-connect/adcc/pull/124`). The new `nuclear_gradient` function either takes a `LazyMp` object as input to compute the gradient for the MP ground state or an `Excitation` object for excited state gradients. A key feature of `adcc` is that workflow extensions beyond the default `run_adc` function are easily accomplished.

As a matter of fact, the computational procedure carried out by `adcc` is rather complex: Depending on the input parameters and selected method, many code execution paths are imaginable, and each combination of parameters requires its own specific subset of pre-computed data. In addition, most quantities are computationally expensive such that they should be computed once and then cached for future purposes. To accomplish this, `adcc` obeys a “demand-driven” strategy, i.e., instead of defining which quantities are needed *a priori*, the computation and caching of a quantity is triggered by the first demand.[80] When first initialized, the objects like `ReferenceState` or `LazyMp` are virtually empty. Neither the integrals are imported yet, nor correlation energies or amplitudes are computed. Only at the point where, e.g., the $\langle \mathsf{oo}||\mathsf{vv}\rangle$ integral block is requested for the first time, its computation and import to `adcc` is triggered. At the next point the same quantity is asked for, existence in the cache is checked, and the object is simply returned if available. The demand-driven approach is easy to implement using standard Python language components like decorators or simple `if` statements. This brings two important advantages. First, the code responsible for computing a single quantity is self-contained, which lowers the code complexity. The working equation is moreover uncoupled from the caching procedure if the aforementioned building blocks are used. Second, the schedule logic and computational algorithm cannot get out of sync. Thus, a major advantage of the demand-driven computation is that it is impossible to forget the pre-computation of a quantity, causing errors at some point. As a result, code entities are orthogonal to each other and quantities become available when needed by `adcc` or user code, not through some pre-defined caching policy, requiring an enormous amount of code adaptations when adding or removing cached results.[80] From a performance perspective, the demand-driven strategy only steadily increases memory usage and lets the calculation run out of memory as late as possible. More details and an illustration of the demand-driven `adcc` workflow can be found in the `adcc` publication.[80]

To enable both flexibility and performance at the same time, `adcc` is implemented using a hybrid Python/C++ approach. The code structure of the library, including main components, data containers, and functions is shown in Figure 4.7. On the lowest level of `adcc`, internally called the `libadcc` library or Core in the illustration, the main C++ building blocks are found. Most importantly, an abstract and convenient tensor interface via the `Tensor` class supporting block-sparse symmetry via `Symmetry` is implemented. The correct block tiling setup is handled by `MoSpaces`, which is required to build new `Tensor` objects from scratch. The actual implementation of `Tensor` is interfaced to the `libtensor` library,[108] a parallelized block-sparse tensor algebra library also used for the same purpose in the original `adcman` implementation[34] in Q-Chem.[93,94] Raw tensor functions like addition, element-wise multiplication, direct summation, or tensor dot products are made available on this low-level already. As of recently, `adcc` supports lazy tensor evaluations through a so-called expression tree which collects a set of tensor operations which are only evaluated once required. This very important feature was implemented by M. F. Herbst and goes hand in hand with the demand-driven philosophy of `adcc` explained above. Moving upward to the Interface layer, a mix of Python and C++ is used to implement the respective functions and classes. The C++ part of the code is conveniently made available on the Python layer via pybind11,[98] where it is amended to yield the full functionality. Most of the performance-critical code is implemented us-

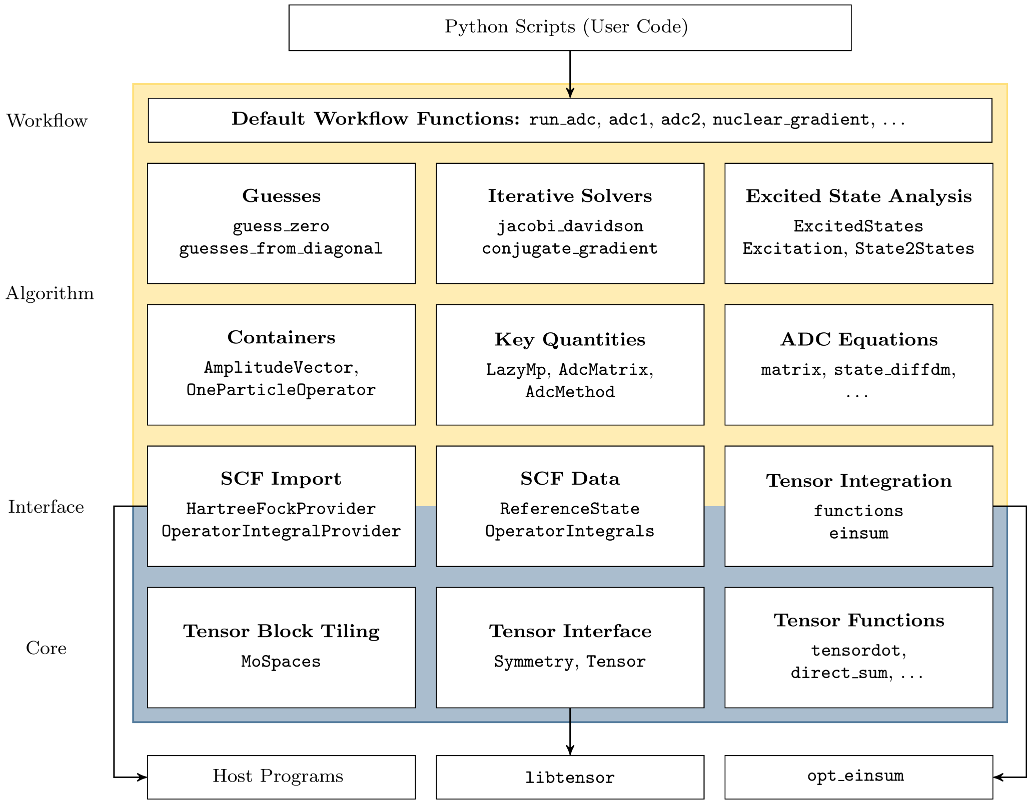


Figure 4.7: Code structure and main components of the `adcc` toolkit. The library is implemented using a mix of Python (yellow) and C++ (blue) code, and it makes use of third-party host programs, the `libtensor` library,[108] and `opt_einsum`.[114] Updated version of the graphic from Ref. 80.

ing C++, i.e., the raw functions to import ERIs, which require a lot of index mapping and bookkeeping. On the contrary, the SCF import is made possible by subclassing, e.g., the `HartreeFockProvider` on the Python layer to import data from third-party host programs. The main SCF data containers, i.e., `ReferenceState` and `OperatorIntegrals` also make use of the mixed implementation. The `OperatorIntegrals` object as a member of `ReferenceState` provides functionality for importing (one-particle) operators from the host program, i.e., the electric dipole operator or the PE induction operator, as we will see later. Tensor integration through convenient and pythonic functions, i.e., operator overloading is mandatory to implement all working equations in Python. These are for example found in the `adcc.functions` file and in the Python-side `Tensor` class. To conveniently express complex tensor contractions for ADC, `adcc` implements an `einsum` function, i.e., tensor contraction expressions using the Einstein summation convention with labels of tensor axes. The tensor contraction order is optimized through the third-party `opt_einsum` library.[114] Of note, the expression tree functionality is fully supported on the Python layer. All the updates to the tensor functionalities since the publication of the `adcc` paper were implemented by M. F. Herbst. Above the Interface layer of `adcc`, the Algorithm layer consists of pure Python code. On the lower level of the Algorithm layer, a tensor container for excitation vectors, the `AmplitudeVector`, is located. The `AmplitudeVector` is nothing but a Python `dict` that makes it possible to store tensors for the individual blocks of the excitation vector, e.g., `ph` and `pphh`. Through some Python magic, algebraic operations on `AmplitudeVector` are forwarded to the individual tensor blocks. A typical construction of an `AmplitudeVector` with tensors `ten1` for `ph` and `ten2` for `pphh` looks like `AmplitudeVector(ph=ten1, pphh=ten2)`. For computation of properties and density matrices, the `OneParticleOperator` provides a multi-tensor container for the respective canonical blocks, i.e., `oo`, `ov`, `vo`, and `vv`. All one-particle operators that are for example imported using `OperatorIntegrals` are ultimately instances of `OneParticleOperator`, which also has convenient short-cuts to the blocks, and expectation values can be computed via the `product_trace` function. As previously explained, the key quantum chemical quantities for ADC are available through `LazyMp` and `AdcMatrix`. The MP working equations are directly implemented inside the `LazyMp` class in Python, whereas the more complicated and long ADC working equations are implemented separately inside the `adc_pp` subdirectory. Therein, the matrix working equations, transition and state densities and modified transition moments are found. Using `adcc.einsum`, the implementation of the working equations is human-readable and can be quickly tested and modified without any code compilation. This feature has brought the rapid prototyping in `adcc` to a whole new level since the first publication. On the upper Algorithm layer, interfaces to guess functions for the iterative solver are implemented, together with the iterative solver schemes themselves. Currently, a Jacobi-Davidson (`jacobi_davidson`) solver is used for eigenvalue problems, whereas a CG solver is available for systems of linear equations, e.g., the orbital response equations for analytical gradients. For the `adcc` users, the excited state containers for analysis are crucial: There are currently three different entities to serve this purpose. First, `ExcitedStates` wraps the entire set of excited states obtained from the eigensolver. Properties are evaluated on demand, by calling the necessary ISR working equations under the hood. The `Excitation` class contains the results of a *single* excited state, whereas the `State2States` can be employed for transitions between excited state, i.e., to model excited state absorption. The `ExcitedStates` class makes use of the rich Python ecosystem for analysis jobs: Plotting of a spectrum with `matplotlib`[109] can be realized with a single line of code by calling `plot_spectrum()` on an `ExcitedStates` object. In addition to excitation spectra, one can, e.g., plot rotatory strengths through `plot_spectrum(yaxis="rotational_strength")`. Despite this default simplicity, the resulting plot is fully customizable through `matplotlib`. As shown above in Figure 4.6, default workflows, handling all building blocks in the right manner for a specific method, are available on the Workflow layer and can be called from user code directly. With the library structure at hand, it becomes clear that `adcc` finds the right balance between customizability and simplicity: A new user is not required to learn everything about the lower-level layers of `adcc`, but can just use the Workflow layer to run their computation. If desired, one can advance to implement more complicated workflows in user code or even implement new working equations. The fourth design goal to only have low barriers for new developers is clearly fulfilled, and has proven very helpful and satisfactory in practice when, for example, students had to carry out a simple programming task with `adcc`.

4.2.3 Environment Models

In this section, I describe my implementation of combined PE-ADC methodologies (Chapter 3) for excitation energies. The presented building blocks of `adcc` allow for straightforward integration of environment models. Let us first recap the computational tasks that need to be carried out to combine PE with ADC. In the perturbative treatment, state difference densities and transition densities are required (see eqs (3.6) and (3.7)) for the ptSS and ptLR term, respectively. These densities are available in the `Excitation` class for an individual excited state. The PE energies, however, need to be evaluated by the host program. Consequently, a natural way to integrate the correction is by exposing possible energy corrections through a specific `HartreeFockProvider` implementation. This is demonstrated with the example of PySCF in Listing 4.6 for both ptSS and ptLR. In line 1, a function to compute the PE induction energy based on an input density is defined. Inside this function, the `get_pe_contribution` routine of the CPPE host program interface of PySCF is called (line 3) and returns the energy as `e_pe`. The specific `HartreeFockProvider` carries a property function that collects all available energy corrections as a Python `dict` (line 21). The `dict` contains the energy correction terms as instances of `EnergyCorrection`, a small class with a name for the energy correction and a Python `callable` to evaluate the correction given an input `Excitation` instance, here named `view`. For each correction, a separate instance of `EnergyCorrection` is created. The `EnergyCorrection` class is designed such that it can be literally “added” to an `ExcitedStates` object by operator overloading. This means that the `__add__` function of `ExcitedStates` is capable of adding `EnergyCorrections` to the excitation energies automatically. The resulting code in the workflow integration reads as natural Python

```python
def pe_energy(self, dm, elec_only=True):
    pe_state = self.scfres.with_solvent
    e_pe, _ = pe_state.kernel(dm.to_ndarray(), elec_only=elec_only)
    return e_pe

@property
def excitation_energy_corrections(self):
    ret = []
    if self.environment == "pe":
        ptlr = EnergyCorrection(
            "pe_ptlr_correction",
            lambda view: 2.0 * self.pe_energy(view.transition_dm_ao,
                                              elec_only=True)
        )
        ptss = EnergyCorrection(
            "pe_ptss_correction",
            lambda view: self.pe_energy(view.state_diffdm_ao,
                                        elec_only=True)
        )
        ret.extend([ptlr, ptss])
    return {ec.name: ec for ec in ret}
```

Listing 4.6: Code excerpt from PySCF `HartreeFockProvider` to set up ptSS and ptLR corrections using `EnergyCorrection`.

code. The `EnergyCorrection` design thus allows for highly customizable inclusion of correction terms. Next, let us turn the attention to the LR-PE formalism for ADC. In eq (3.17), the ADC matrix was modified to include a PE-specific response term. The corresponding matrix-vector product is shown in eq (3.13). As such, an extra term needs to be added to the lazy matrix representation `AdcMatrix`. For this reason, I created the `AdcExtraTerm` class, which contains a reference to the original matrix instance (for accessibility of the `ReferenceState` and `LazyMp` ground state) and a dictionary of block spaces and callables for the extra term. For LR-PE-ADC, I have shown that only a contribution to the `ph-ph` block is added. The code to compute this contribution from eq (3.13) is shown in Listing 4.7. The resulting `AdcBlock` instance is an `adcc` data structure responsible for bookkeeping matrix apply and diagonal routines, also for the main working equations. The `AdcBlock` instance for the LR-PE matrix-vector product contribution can be assigned to an `AdcExtraTerm` and added to `AdcMatrix`. This mechanism is again handled by overloading the `__add__` operator of `AdcMatrix`. As a result, the manual addition of an extra term `term` to a matrix `matrix` looks like `matrix += term`. In contrast to the implementation of energy corrections, obtaining the PE induction opera-

```
1 def block_ph_ph_0_pe(hf, mp, intermediates):
2     op = hf.operators
3     def apply(ampl):
4         tdm = OneParticleOperator(mp, is_symmetric=False)
5         tdm.vo = ampl.ph.transpose()
6         vpe = op.pe_induction_elec(tdm)
7         return AmplitudeVector(ph=vpe.ov)
8     return AdcBlock(apply, 0)
```

Listing 4.7: Code to compute the LR-PE contribution to the `ph-ph` block of the ADC matrix-vector product (eq (3.13)), contained in file `adc_pp/environment.py`.

tor is a bit more involved. This task was, however, easily achieved relying on the `adcc` design philosophy and existing library infrastructure. I have simply added an operator to the top-level `OperatorIntegrals`, namely `pe_induction_elec` and its counterpart to the program-specific `OperatorIntegralProvider` implementation. The `pe_induction_elec` property returns a function that 1) takes an input density, 2) transforms the density to the AO basis, 3) computes the PE induction operator in the host program, and 4) transforms the induction operator back to the MO basis. The corresponding routine is called `__import_density_dependent_operator` inside the `OperatorIntegrals` class. Step 3) is implemented in the program-specific `OperatorIntegralProvider` as illustrated for PySCF in Listing 4.8. All of this happens behind the scenes in line 6 of Listing 4.7, and the programmer does not need to take care of it manually. Adding similar environment models should be easy, since the machinery to do so is already in place. The integration of the PE induction operator through `OperatorIntegrals` also makes it possible to implement all ADC-related routines for analytic gradients, i.e., orbital response con-

```
@property
def pe_induction_elec(self):
    if hasattr(self.scfres, "with_solvent"):
        if isinstance(self.scfres.with_solvent, solvent.pol_embed.PolEmbed):
            def pe_induction_elec_ao(dm):
                return self.scfres.with_solvent._exec_cppe(
                    dm.to_ndarray(), elec_only=True
                )[1]
            return pe_induction_elec_ao
```

Listing 4.8: Code to set up the callable for the PE electronic induction operator in `PyscfOperatorIntegralProvider`.

tributions. To summarize, perturbative corrections for pt-PE-ADC are implemented via the `EnergyCorrection` class, whereas modification of the matrix-vector product for LR-PE-ADC is enabled through `AdcExtraTerm` added to an `AdcMatrix` instance. No further changes to the solver algorithm or any other part of the code are required due to the design of `adcc`: Only the `AdcMatrix` needs to know which contributions are required. This means that for implementing PE-ADC response properties, no further work is needed. This clearly emphasizes how cleanly the `adcc` interfaces and building blocks are designed, because also more complex additions like LR-PE-ADC could easily be added. Without `adcc`, the implementation of said methods would have taken significantly longer, both in terms of implementation and debugging the code.

Integration into the standard `adcc` workflow

For the `adcc` end user, assembling all the building blocks for PE-ADC calculations would be tedious, inefficient, and error-prone. Therefore, environment models are natively integrated in the standard `adcc` workflow outlined above. I have added an environment parameter `environment` to the `run_adc` function. Then, inside `run_adc`, after the ADC matrix has been set up, the `setup_environment` routine is called with the `matrix` and `environment` parameter (see Listing 4.9). The content of the `setup_environment` function will be explained below. It returns the additional matrix term `env_matrix_term` as an `AdcExtraTerm` instance, if no additional term is required, the variable is set to `None`. In line 6 of Listing 4.9, the additional matrix term is added to the `AdcMatrix`

```
1  # Setup environment coupling terms and energy corrections
2  ret = setup_environment(matrix, environment)
3  env_matrix_term, env_energy_corrections = ret
4  # add terms to matrix
5  if env_matrix_term:
6      matrix += env_matrix_term
7
8  # ...
9  # [solve ADC eigenvalue problem and get ExcitedStates exstates]
10 # ...
11
12 # add environment corrections to excited states
13 exstates += env_energy_corrections
```

Listing 4.9: Code excerpt from `adcc.run_adc` responsible for the addition of environment models, i.e., matrix terms and energy corrections.

instance. For perturbative corrections, the returned `env_energy_corrections` is a `list` of `EnergyCorrections`, added to the `ExcitedStates` object in the last line. Of course all the actual setup takes place inside `setup_environment`. The function is too long to be displayed here, but the key steps are given in the following. First, the function checks whether an environment is actually present for the given `ReferenceState` by checking its `environment` property. If `environment` is set to `None` but an environment is found for the reference state, an error is thrown. Also in the case where environment parameters are specified, but no environment is present, the function throws an error. This ensures that the user is actually aware of what should happen during the `adcc` calculation. Second, the provided `environment` parameter is translated into a consistent and valid Python `dict`. Various possibilities exist to specify `environment` to grant some flexibility. For example, one can set `environment=False` (boolean argument) to request a PTE-PE-ADC calculation, i.e., no extra terms or corrections are added. To get pt-PE-ADC with both ptSS and ptLR corrections, `environment=["ptss", "ptlr"]` is the parameter to set, or `environment=True`. These two inputs are converted to `{"ptss": True, "ptlr": True}` which is the final form of the sanitized input parameter. If invalid arguments or combinations are given, an error is returned. Third and last, the requested terms are prepared. The code checks if the requested perturbative correction is present in the `excitation_energy_corrections` field of `ReferenceState` and collects all corrections in a `list`. In case of LR-PE-ADC, available via `environment="linear_response"`, the corresponding `AdcExtraTerm` is created from the working equation in `adc_pp/environment.py` (Listing 4.7) and returned as `env_matrix_term`. More details on the options to be set in practice can be found in the `adcc` online documentation. With the seamless integration of the PE-ADC in the standard `adcc` workflow routines, the entire machinery to calculate analytic gradients with PE-MP or PE-ADC is ready.

4.2.4 Analytic Gradients

Before turning myself to analytic gradients with PE, I had to implement the entire set of vacuum MP and ADC gradients in `adcc`, since these features were not yet available. The gradient features are not released yet and under development (see `https://github.com/adc-connect/adcc/pull/124`), consequently, the implementation details presented in the following might be subject to changes in the feature. Keeping the PE contributions aside for a moment, the canonical flow I have implemented for analytical gradients in `adcc` is outlined in Figure 4.8. It consists of a main function, `nuclear_gradient`, which either takes a `LazyMp` object for MP gradients or an `Excitation`

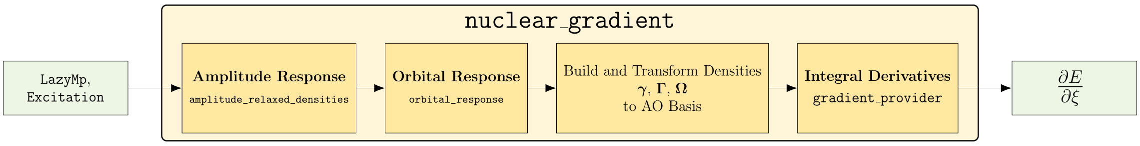


Figure 4.8: Illustration of the `nuclear_gradient` function of `adcc`. Based on a `LazyMp` or `Excitation` input, the function assembles all contributions to obtain effective densities through amplitude and orbital response and triggers evaluation of the final gradient through contraction with integral derivatives in the host program.

object for ADC gradients as input parameter. Inside the function, the work to be carried out is split into four parts. As a first step, the amplitude-relaxed densities of the specified wave function method are computed. All working equations for the vacuum gradients were taken from Refs. 76 and 78. The `amplitude_relaxed_densities` function, contained in `gradients/amplitude_response.py`, dispatches to the correct working equations for the amplitude-relaxed densities for the given input. One-particle density matrices are naturally stored in `OneParticleOperator` objects, whereas I have created a `TwoParticleDensityMatrix` class for handling two-particle densities conveniently (short-cut block access, addition of two densities, MO-to-AO transformation, etc.). The amplitude-relaxed densities are subsequently used to obtain all orbital response Lagrange multipliers $\{\lambda_{pq}\}$ and $\{\omega_{pq}\}$. First, the right-hand side for the iterative solution of λ_{ia} is computed (`orbital_response_rhs`). Second, the right-hand side is passed to the `orbital_response` routine, where the multipliers are iteratively solved with the CG algorithm of `adcc`. Third, the $\{\omega_{pq}\}$ multipliers are evaluated. For contraction with integral derivatives, all one- and two-particle density matrices are built and then transformed to the AO basis. For the final step, another new component developed according to `adcc`'s design philosophy comes into play: each `HartreeFockProvider` that should be used for gradient evaluations passes its own `GradientProvider` implementation on to the `ReferenceState`. The task carried out by the host-program-specific `GradientProvider` is to take input one- and two-particle density matrices and contract those with the integral derivatives from the host program directly. In this manner, no differentiated two-electron integrals need to be imported into `adcc`, which would be computationally rather expensive. Through the clean `GradientProvider` interface, however, the correct backend is automatically chosen, such that all host-program-specific code stays in a single place. Currently, MP and ADC gradients are available with the Psi4 and PySCF backends.

As shown in Section 3.4, PE contributions arise in several spots of the analytic gradient equations. The amendments to be made for PE orbital response both for the PTE and the

LR scheme were shown in Table 3.1. These needed to be injected in several spots in the vacuum implementation. The equation for the orbital response right-hand side k_{ia} for PE is simply `-2.0 * hf.operators.pe_induction_elec(g1a).ov`, where `g1a` is the amplitude-relaxed one-particle density matrix. This term is conditionally added if PE is present in the reference state. The orbital response iterations contain an additional PE term in the orbital Hessian matrix `OrbitalResponseMatrix`. Since these response equations are iteratively solved, only the matrix-vector product function `__matmul__` needs to be implemented, shown in Listing 4.10. The vacuum part is implemented in lines 2 to 7, and the extra term is added if PE is present, again using the `pe_induction_elec` functionality of `OperatorIntegrals`. Adding the PTE-PE contribution to the ω_{ij} multipliers is straightforward. Furthermore, the contributions to the orbital response that arise from the LR-PE term in the ADC matrix are constant terms to be added to k_{ia} and the three canonical blocks of $\{\omega_{pq}\}$, feasible in a couple of code lines. In the above code examples, the ease of implementation using `adcc` is outstanding, because the equations are still clearly legible, like in a textbook, and the concise integration of PE does not disturb the code readability. Generally, the `nuclear_gradient` is rather concise with only approximately 100 lines of code. As such, the pilot workflow for analytical gradients in `adcc` could be implemented with the same philosophy as the rest of the library, making available new building blocks that are easily assembled for a specific, pre-defined feature. Last but not least, the effective one- and two-particle density matrices obtained from the orbital response also need to be contracted with PE integral derivatives as in eqs (3.73) and (3.79). This task is han-

```
1  def __matmul__(self, l_ov):
2      ret = (
3          + einsum("ab,ib->ia", self.hf.fvv, l_ov)
4          - einsum("ij,ja->ia", self.hf.foo, l_ov)
5          + einsum("ijab,jb->ia", self.hf.oovv, l_ov)
6          - einsum("ibja,jb->ia", self.hf.ovov, l_ov)
7      )
8      if self.hf.environment == "pe":
9          ops = self.hf.operators
10         dm = OneParticleOperator(self.hf, is_symmetric=True)
11         dm.ov = l_ov
12         ret += ops.pe_induction_elec(dm).ov
13     return evaluate(ret)
```

Listing 4.10: Python implementation of the orbital Hessian matrix-vector product inside the `OrbitalResponseMatrix` class from eq (3.43). Here, `l_ov` are the input λ_{jb} multipliers.

dled by the host-program-specific `GradientProvider` implementation and the explicit PE contributions are then added to the final gradient. The analytic gradient implementation (both vacuum and with PE) was tested against numerically obtained nuclear gradients with a five-point stencil finite difference scheme. These tests are also added to the `adcc` test framework run through the CI.

4.2.5 Current State and Code Contributions

The core feature set of `adcc` was described in the publication,[80] and, regarding the plain ADC functionality, has remained mostly the same. The library was, however, made fully available as an open-source package on GitHub, where the entire development now takes place. In addition, all working equations were transferred to the Python layer through the new tensor integration features. This largely facilitates implementation of low-level new features, like the gradients mentioned above. Currently, gradients for MP2, ADC(1), and ADC(2) are implemented in the development branch of `adcc`, with and without using PE as environment model. Furthermore, the PE formalisms pt-PE-ADC and LR-PE-ADC were recently added, in conjunction with the transparent `environment` parameter handling. Over the last two years, `adcc` was equipped with more and more features, including state-of-the-art environment models which are missing in established program packages. This was made possible by the entire design of `adcc`, as envisioned in the design goals. In the following, I present a breakdown of the most important features I myself have contributed to `adcc`:

- Implementation of `HartreeFockProvider` subclasses for Psi4, PySCF, and VeloxChem, including support for unrestricted references (e.g., PR #20).
- Operator import through `OperatorIntegrals` and `OperatorIntegralProvider`.
- Maintenance of automatic deployment to Anaconda.
- Python-side `OneParticleOperator` implementation (PR #119).
- Python-side `LazyMp` implementation (PR #104).
- Rotatory strengths for simulation of ECD spectra (PR #56).
- Export of `ExcitedStates` to `pandas.DataFrame` (PR #78).
- State-to-state transition density matrices (`State2States`, PR #91).
- Environment model interface for pt-PE and LR-PE methodologies (PR #123).

- Analytic nuclear gradients for MP2, ADC(1), and ADC(2) (PR #124).
- Analytic nuclear gradients for PE-MP2, PE-ADC(1), and PE-ADC(2) (`pe_gradients` branch).

Numbers in parentheses indicate the corresponding pull request (PR) in the `adcc` GitHub repository. The last two points are not yet merged to the main `adcc` branch, but fully functional. The SCF interfaces had been implemented before `adcc` went open-source.

4.3 Respondo: Library for ADC/ISR Response Functions

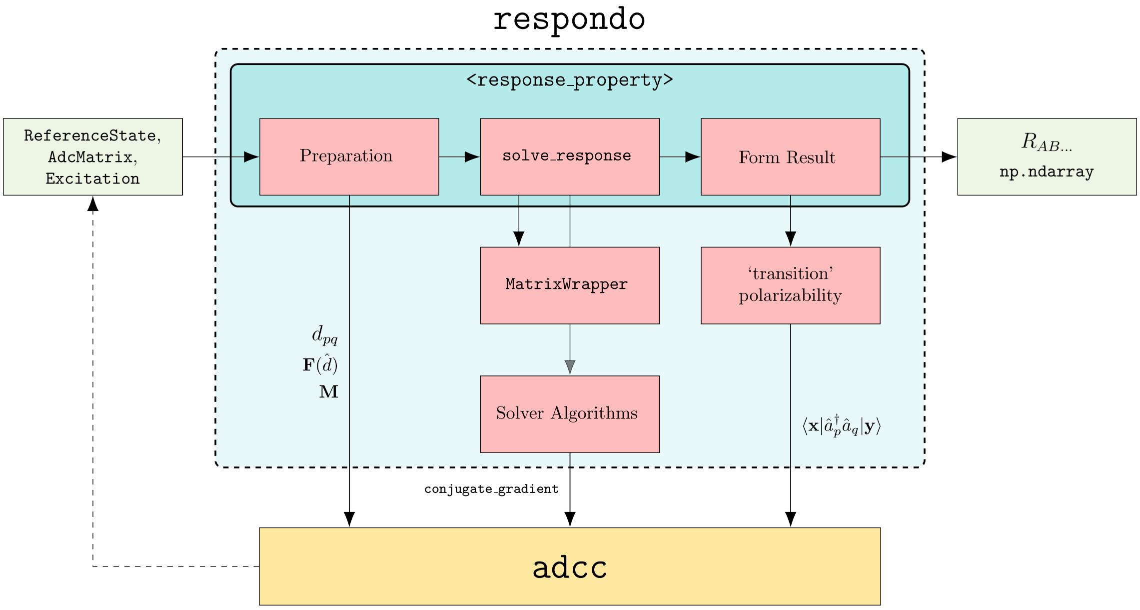


Figure 4.9: Overview of `respondo`. The program flow consists of calling a compute routine for an implemented response property `<response_property>`, which first prepares the solution of linear equations, then solves these equations and forms the output tensor. Red boxes are components of `respondo`, and the ADC/ISR quantities obtained from `adcc` are indicated next to the respective arrows.

The last component needed to implement all previously presented PE-ADC methodologies is a library for computing response functions based on ADC. For this purpose, I implemented the `respondo` library as part of the Gator project.[84] The `respondo` library can be viewed as a small plugin for `adcc` which implements the necessary workflow for ADC/ISR response properties. In Section 2.4.4, I have shown the fundamentals of iteratively solving ADC/ISR response equations, and `respondo` picks up on these theoretical foundations. The structure and workflow of `respondo` is schematically illustrated in Figure 4.9. The library fully relies on `adcc`-provided data structures, such as `ReferenceState`, `AdcMatrix`, or `Excitation` as input objects. Each response property implemented in `respondo` has a specific compute function, here denoted generically as `<response_property>`. I will explain the individual steps with the example of the complex ground state polarizability, Based on the input data and the property at hand, solution of the response equations is first prepared by retrieving the appropriate operators, modified transition moments, and ADC matrix representation from `adcc`. The respective code for the complex polarizability is shown in Listing 4.11. The right-hand side of the linear equations are here

```
def complex_polarizability(
    data_or_matrix, method=None, omega=0.0, gamma=0.0, **solver_args
):
    # obtain matrix
    matrix = construct_adcmatrix(data_or_matrix, method=method)
    property_method = select_property_method(matrix)
    hf = matrix.reference_state
    mp = matrix.ground_state
    dips = hf.operators.electric_dipole
    # compute right-hand side
    rhss = modified_transition_moments(property_method, mp, dips)
```

Listing 4.11: Preparation part of the `complex_polarizability` function in `respondo`.

equal to the modified transition moments computed with the electric dipole operators `dips`. In the following step, the property routine calls the `solve_response` function, a convenient black-box routine to solve a linear equation based on a given matrix, right-hand side, and frequency arguments. The code block calling `solve_response` is shown in Listing 4.12. Under the hood, distinct code paths have to be chosen based on the selected solver algorithm. Currently, two different solver algorithms are used, namely the new CPP solver presented in Chapter 7 and the CG solver from `adcc`. In addition, a specific variant of the ADC(2) matrix can be used, where the `pphh` part of the response vector is folded into the `ph` part. Details will be explained in Chapter 7. To conveniently wrap all kinds of different matrix representations needed in `respondo`, I designed the `MatrixWrapper` class, whose structure is shown in Figure 4.10. Here, it is most important that the `MatrixWrapper` class provides a *single* interface to the matrix-vector product via its `__matmul__` function, together with helper routines to adapt to different solvers and matrix representations. Once the `MatrixWrapper` instance is created, `solve_response` calls the desired iterative solver algorithm and returns the solution vector. The final part of the `complex_polarizability` function is presented in Listing 4.13. In this part, an empty

```
response_positive = [
    solve_response(matrix, ResponseVector(rhs),
                   omega, gamma, **solver_args)
    for rhs in rhss
]
if omega == 0.0:
    response_negative = response_positive
else:
    response_negative = [
        solve_response(matrix, ResponseVector(rhs),
                       -omega, gamma, **solver_args)
        for rhs in rhss
    ]
```

Listing 4.12: Solving part of the `complex_polarizability` function in `respondo`, calling `solve_response` for each Cartesian component of the right-hand side.

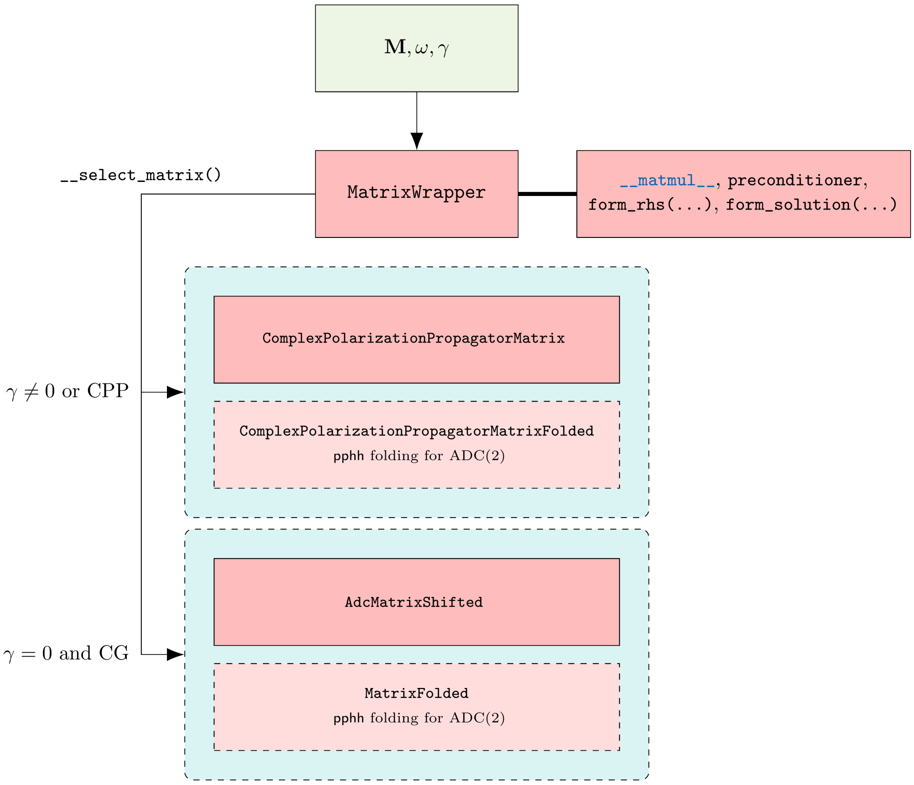


Figure 4.10: Overview of `MatrixWrapper` with different matrix representations wrapped and exposed via a single algebra interface.

complex tensor is created, which is then filled by computing scalar products of modified

```
polarizability = np.zeros((3, 3), dtype=np.complex)
for A in range(3):
    for B in range(A, 3):
        rsp_pos = response_positive[B]
        rsp_neg = response_negative[B]
        polarizability.real[A, B] = (
            rsp_pos.real @ rhss[A] + rsp_neg.real @ rhss[A]
        )
        polarizability.imag[A, B] = (
            rsp_pos.imag @ rhss[A] - rsp_neg.imag @ rhss[A]
        )
        polarizability[B, A] = polarizability[A, B]
return polarizability
```

Listing 4.13: Construction of the output polarizability tensor inside the `complex_polarizability` function.

transition moments and response vectors (see Table 2.2). For other properties, transition densities between two vectors have to be obtained (e.g., TPA or RIXS) because there is currently no implementation of the operator matrix **B** available in `adcc`. This task is handled by the `transition_polarizability` routine. Finally, the property routine returns the output tensor as a NumPy array `np.ndarray`. All properties implemented in `respondo` are tested by carrying out a full ADC matrix diagonalization and then explicitly evaluating the corresponding SOS terms. The results from the SOS and the iterative solution are checked for equality.

At the current stage, `respondo` is capable of computing a variety of response properties, summarized in Table 4.1. Note that *all* of these properties are also supported together

Table 4.1: Response properties currently supported in `respondo`.

Property	Function	Equation
Static polarizability	`static_polarizability`	(2.78), $\omega = 0,\ \gamma = 0$
Frequency-dependent polarizability	`real_polarizability`	(2.78), $\gamma = 0$
Complex polarizability	`complex_polarizability`	(2.78)
RIXS scattering amplitude	`rixs`	(2.91)
TPA matrix	`tpa_resonant`	(2.86)

with LR-PE, as explained in Section 3.3: The only thing that needs to be taken care of is the inclusion of the additional matrix term through the `AdcMatrix`. This has already been achieved during the LR-PE implementation in `adcc`. Consequently, no PE-specific code enters the `respondo` library. The only thing one needs to do is to use the appropriately modified `AdcMatrix` to perform the matrix inversions. In this respect, the synergy of `adcc` and `respondo` is clearly outstanding, because all properties that will eventually be added to `respondo` will automatically be available for PE-ADC, without further ado.

4.4 Summary

To conclude this chapter, I want to demonstrate how one can run all the previously derived and implemented PE-ADC methodologies for ground state PE, excitation energies, response properties, and gradients in a single Python script, shown in Listing 4.14. In this example, the formaldehyde molecule is used as the quantum region and six water molecules in its environment are modeled through PE. In line 20, the PE-SCF problem in PySCF is created and solved in line 23, making use of CPPE interfaced to PySCF in the background. Then, five singlet excited states with pt-PE-ADC(2) are computed, taking into account ptSS and ptLR terms. For comparison, the same task is carried out but with LR-PE-ADC(2) by just switching the `environment` argument. Both `ExcitedStates` results are printed as a table through the `describe` function. Next, `respondo` comes into play and computes the complex polarizability with the PE-ADC(2) matrix, i.e., including the PE linear response term. In the last line of the input example, the LR-PE-ADC(2) gradient of the first excited state is evaluated. This input example nicely shows how the methodologies derived in Chapter 3 can be applied in practice, exploiting the combination of *all* open-source libraries described in the current chapter – CPPE, `adcc`, and `respondo`. Through these libraries, the entire set of PE-ADC combinations is openly available for everyone, which is a unique feature set for advanced environment modeling. The first computational results with the presented methodologies and the implemented libraries are shown in Chapter 5. In the future, the open-source libraries will be further expanded and improved, especially the gradient implementation with PE-ADC is not capable of carrying out efficient and stable geometry optimizations in the current state. Once this is achieved, ever more complex systems can be studied with the comprehensive feature set provided by the open-source libraries for computational spectroscopy illustrated here.

```python
import adcc
import numpy as np
from pyscf import gto, scf
from pyscf.solvent import PE
from respondo.polarizability import complex_polarizability

mol = gto.M(
    atom="""
    C 1.0632450881806 2.0267971791743 0.4338879750526
    O 1.1154451117032 1.0798728186948 1.1542424552747
    H 1.0944666250874 3.0394904220684 0.8360468907200
    H 0.9836601903170 1.9241779934791 -0.6452234478151
    """,
    basis='sto-3g',
    charge=0,
)
options = {"potfile": "fa_6w.pot"}

# Run PE-SCF with PySCF
scfres = PE(scf.RHF(mol), options)
scfres.conv_tol = 1e-10
scfres.conv_tol_grad = 1e-7
scfres.kernel()
# set up reference state
refstate = adcc.ReferenceState(scfres)

# run pt-PE-ADC(2)
state_pt = adcc.adc2(refstate, n_singlets=5, environment=['ptss', 'ptlr'])
print(state_pt.describe())
# run LR-PE-ADC(2)
state_lr = adcc.adc2(refstate, n_singlets=5, environment='linear_response')
print(state_lr.describe())

# solve complex polarizability at the second excitation energy
omega = state_lr.excitation_energy[1]
alpha = complex_polarizability(state_lr.matrix, omega=omega, gamma=0.0046)

# compute the nuclear gradient of the first excited state
grad = adcc.nuclear_gradient(state_lr.excitations[0])
```

Listing 4.14: Example input file for running PE-ADC calculations with PySCF, CPPE, `adcc`, and `respondo` for formaldehyde.

Notes

1. Reprinted with permission from: M. Scheurer, P. Reinholdt, E. R. Kjellgren, J. M. H. Olsen, A. Dreuw, and J. Kongsted, “CPPE: An Open-Source C++ and Python Library for Polarizable Embedding”, *J. Chem. Theory Comput.* **2019**, *15 (11)*, 6154-6163. Copyright 2019 American Chemical Society.

Chapter 5

Application of PE-ADC Methodologies

The aim of this chapter is to expand upon the theoretical derivations and implementations of combined PE-ADC methodologies via test calculations of electronic excitation energies, higher-order response properties, and analytic nuclear gradients. In the first part, benchmarks and applications of the perturbative PE-ADC combined method are briefly summarized. Then, a first benchmark study of the PE-ADC/ISR response formalism is presented by evaluating the two-photon transition strengths of two solvated chromophores. The correctness of the PE gradient implementations is furthermore verified through a simple numerical test case. In the end, I give a short summary and ideas for future applications on the devised methods.

5.1 Perturbatively Corrected PE-ADC Excitation Energies

The variant of the ADC scheme combined with the PE model for accurate calculations of electronic excited states including perturbative corrections was presented in detail in my publications.[42,61] Therein, I demonstrate the accuracy and large-scale applicability with two benchmark studies and a biomolecular case study, analyzing the importance of both ptSS and ptLR corrections for the excitation energies in presence of polarizable environments.[42,61] In this study, I used my PE-ADC implementation in the Q-Chem program package,[93,94] but the same calculations can also be carried out with `adcc`. In

---

Parts of this chapter have already been published in:

- M. Scheurer, M. F. Herbst, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted, "Polarizable Embedding Combined with the Algebraic Diagrammatic Construction: Tackling Excited States in Biomolecular Systems", *J. Chem. Theory Comput.* **2018**, *14 (9)*, 4870-4883. (Reference 61)
- M. Scheurer, "Polarizable Embedding for the Algebraic-Diagrammatic Construction Scheme", *Springer Fachmedien Wiesbaden*, **2020**. (Reference 42)

the following, I briefly summarize the results of the pt-PE-ADC benchmark computations and explain how I applied the scheme to accurately model a charge transfer (CT) excitation in the active site of the flavoprotein dodecin.

In a first test, the low-lying $\pi \rightarrow \pi^*$ and $\mathrm{n} \rightarrow \pi^*$ singlet states of *para*-nitroaniline (pNA) embedded in small water clusters were computed with pt-PE-ADC(2) and pt-PE-ADC(3/2) are compared to the respective ADC result for the molecular supersystem.[42,61] Since pNA forms hydrogen bonds to the surrounding water molecules via the lone pair of the nitro group, this test system is particularly challenging. The expected blue shift was observed for the $\mathrm{n} \rightarrow \pi^*$ excitation energy in presence of the water clusters through destabilization of the excited state. In this case, ptSS and ptLR corrections were almost equal to zero, because the transition is forbidden and the excited state dipole moment does not change much compared to the ground state. Consequently, the observed blue shift arises only through PE contributions to the ground state wave function. The properties of the $\pi \rightarrow \pi^*$ transition of pNA in the presence of small water clusters are, on the other hand, quite different. This transition corresponds to an intramolecular CT, such that the electric dipole moment approximately doubles upon excitation. In addition, the transition is allowed, such that the corresponding oscillator strength is larger than zero. Both perturbative corrections were non-negligible in this case. The expected red shift of the $\pi \rightarrow \pi^*$ excitation energy was correctly captured through pt-PE-ADC, even though the largest contribution to the solvent shift always stems from the PE contribution to the reference state.[42,61] The maximum absolute errors found for pt-PE-ADC in comparison with supersystem ADC calculations were only 0.07 eV for ADC(2) and 0.06 eV for ADC(3/2),[42,61] well below the intrinsic error of the employed quantum chemical method itself.[35] The second case study analyzes bulk solvation effects on the electronic excited states of lumiflavin (Lf) via pt-PE-ADC(2).[42,61] In this case a supersystem computation was not feasible because of the large number of solvent molecules. Instead, the results were compared to the well established PCM-ADC(2) method using continuum solvation, which also relies on perturbative corrections.[115,116] For pt-PE-ADC(2), configurations of the entire system were sampled with QM/MM MD simulations, whereas PCM-ADC(2) calculations were run on optimized structures of Lf in presence of the solvent continuum. Two different solvents – water and cyclohexane – were employed. The comparison of pt-PE-ADC(2) and PCM-ADC(2) showed good agreement of solvent shifts for $\pi \rightarrow \pi^*$ excitations. On the contrary, different trends were predicted in case of $\mathrm{n} \rightarrow \pi^*$, which might result from the underlying sampled structures in case of pt-PE-ADC(2). A statistical analysis of all snapshots and their excited state properties however showed that the number of snapshots was large enough for meaningful median excitation energies.

The explicit and continuum solvation methods were found to agree in their description of bright states, which are relevant for spectroscopy simulations. An illustration on the general relationship between the magnitudes of the ptSS and ptLR corrections to excited state properties, i.e., difference dipole moment and oscillator strength, respectively, was presented.[42,61]

The last example demonstrates the use case pt-PE-ADC was developed for: Modeling excited states of chromophores embedded in complex environments, such as photoreactive proteins. Therefore, the CT state pivotal for the photoprotection mechanism of the flavoprotein dodecin was analyzed with the combined method.[61,117–119] Dodecin, a homododecameric protein complex, is the key player for flavin homeostasis in archaebacteria.[117] It is capable of storing several types of flavin derivatives and efficiently protects those photolabile species from photodegradation through a multistep quenching mechanism.[119,120] The key step of the quenching mechanism is a CT excitation from an adjacent tryptophan (W36) residue to the flavin species bound by dodecin. Upon CT excitation, charge separation results in a positively charged tryptophan radical cation and a negatively charged flavosemiquinone radical, investigated using *in vacuo* calculations in a previous work of mine.[120] An alternative pathway corresponds to a local excitation (LE) on the flavin species, leading to photodegradation. For this reason, it is clear that dodecin must be capable of efficiently promoting the CT excitation for photoprotection. To gain further understanding whether a polarizable protein environment directly promotes this pathway, the excitation process of the Lf-W36 dimer was elucidated using pt-PE-ADC(2). The computational protocol for MD sampling, QM/MM MD simulations, and quantum chemical calculations is presented in detail in Ref. 120. For the resulting 50 representative snapshots, three singlet excited states were computed with pt-PE-ADC(2),[61] where the lowest excited states were identified as CT transitions. The nature of this transition and relative ordering of the states depend on the specific configuration, i.e., relative orientation of Lf and W36.[61] Sometimes a mixture of LE and CT states is observed. For a CT state, the magnitude of the ptSS correction is expected to be large due to the linear relationship with the difference dipole moment. Indeed, snapshots with a ptSS correction smaller than -0.05 eV for the lowest excited states were identified as CT states using the electron-hole distance $d_{\mathrm{h}\rightarrow\mathrm{e}}$ as a direct probe for CT character. In Figure 5.1a, the direct relationship between the magnitude of the ptSS correction and the electron-hole distance is illustrated for the entire set of snapshots. The larger the electron-hole distance, the larger the CT character of the excitation, and consequently, the ptSS correction exhibits a large magnitude. The effect of the polarizable environment in these computations is illustrated in Figure 5.1b and c: With PE enabled, the first three singlet states show significant CT

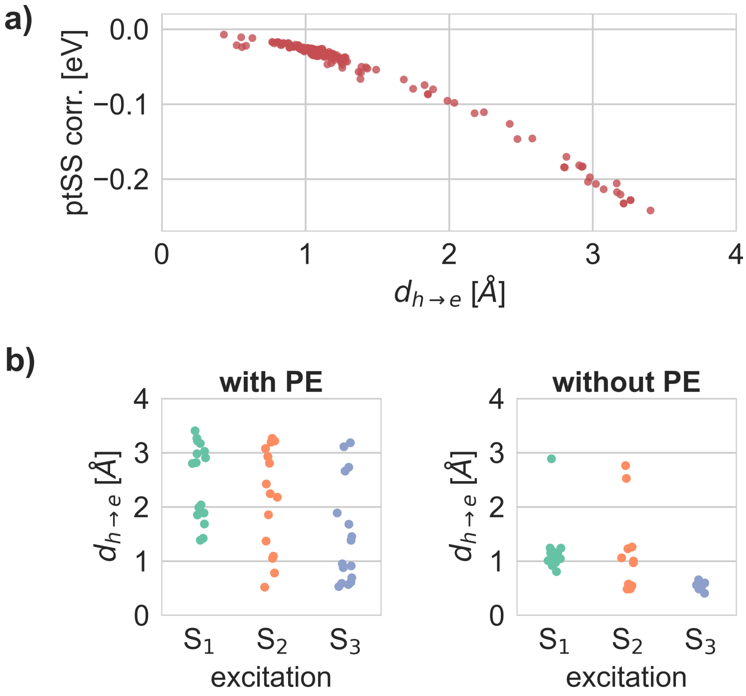


Figure 5.1: Analysis of Lf-W36 CT states using electron-hole distances. **a)** Relationship between ptSS correction and electron-hole distance $d_{\mathrm{h}\rightarrow\mathrm{e}}$ for S_1 and S_2. **b)** Effect of PE on electron-hole distances. In isolation, i.e., without the protein environment, the character of the excitation is shifted from CT to LE.[1]

character on average. This observation changes dramatically when the polarizable environment is not present anymore. In this case, the LE state dominates, and almost no CT character is present anymore for the Lf-W36 dimer. Clearly, the polarizable dodecin and water environment stabilize the CT excitation of the Lf-W36 dimer. Of course, the nature of the excited state is not changed through the perturbative corrections, but only through the PE contribution to the reference state. In order to describe photoexcitations in such flavoproteins realistically, a sophisticated environment model should be chosen rather than ignoring the pivotal influence of the polarizable environment. As a next step, the manipulation of the dodecin photocycle should be studied using pt-PE-ADC to unravel the influence of redox potentials of the involved tryptophan derivative.[119]

5.2 Response Properties with PE-ADC

This section illustrates the PE-ADC/ISR response approach as derived in Section 3.3 applied to two-photon absorption simulations of small molecules. The implementation in `adcc` and `respondo` was described in the previous chapter. Recall that in order to evaluate a PE-ADC/ISR response property, only the ADC matrix has been modified to include a zeroth-order coupling term to the polarizable environment in an iterative manner. In the following, I assess the accuracy of the PE-ADC approaches by comparing to supersystem computations of excitation energies, oscillator strengths, and TP absorption strengths δ_{TP}. To obtain accurate results, two additional corrections are employed. First, electron spill-out, i.e., delocalization of the electron density to the environment is avoided via the PE(ECP) approach.[106] These simple pseudopotentials, located at the environment expansion sites, capture the missing Pauli repulsion between the quantum region and the environment, such that artifacts from electron spill-out are minimized.[32] Other approaches to avoid electron spill-out in PE calculations exist,[105] however, the PE(ECP) scheme is much easier to implement. It was shown in previous work that PE(ECP) effectively remedies electron spill-out in PE-ADC computations,[106] and similar results were shown for TPA simulations with TDDFT recently.[121] Inclusion of repulsive potentials is almost mandatory with anionic species in the quantum region or diffuse basis sets. The PE(ECP) potential is a simple one-electron contribution added to the core Hamiltonian during the SCF, and is thus independent of the post-SCF treatment. As explained previously, I added the PE(ECP) option to PySCF and Psi4. A second improvement for molecular response properties is the inclusion of local electric field effects, called effective external fields (EEF),[122] in order to obtain correct transition moments. Without EEF, the transition moments computed from the electric dipole operator completely ignore the transition induced moments created in the polarizable environment. This physically wrong behavior can be corrected by modification of the electric dipole operator matrix elements. The effective dipole operator is then used to compute, e.g., transition dipole moments or modified transition moments as right-hand side for response equations.[122] Using EEF is a standard procedure in PE-TDDFT simulations, however, it has never been used together with PE-ADC until now. To discern the effects of the coupling scheme, PE(ECP), and EEF, different combinations of these settings are tested in the following.

The molecular structure of the deprotonated anionic chromophore of photoactive yellow protein (PYPb), with two water molecules located next to the phenolate oxygen atom, was taken from previous work.[70,123] The system is referred to as PYPb/2 H_2O in the following, depicted in Figure 5.2a. Embedding parameters, including static multipole moments (charges, dipoles, quadrupoles) and dipole-dipole polarizabilities were obtained with the LoProp approach[124] using PyFraME.[125] The LoProp calculations were run using Dalton[88,126] with CAM-B3LYP and the loprop-6-31+G* basis set.[127] For the following ADC calculations, PySCF was used as the host program in combination with `adcc`. Three excited singlet states were obtained at the ADC(2)/6-31+G* level of theory for the

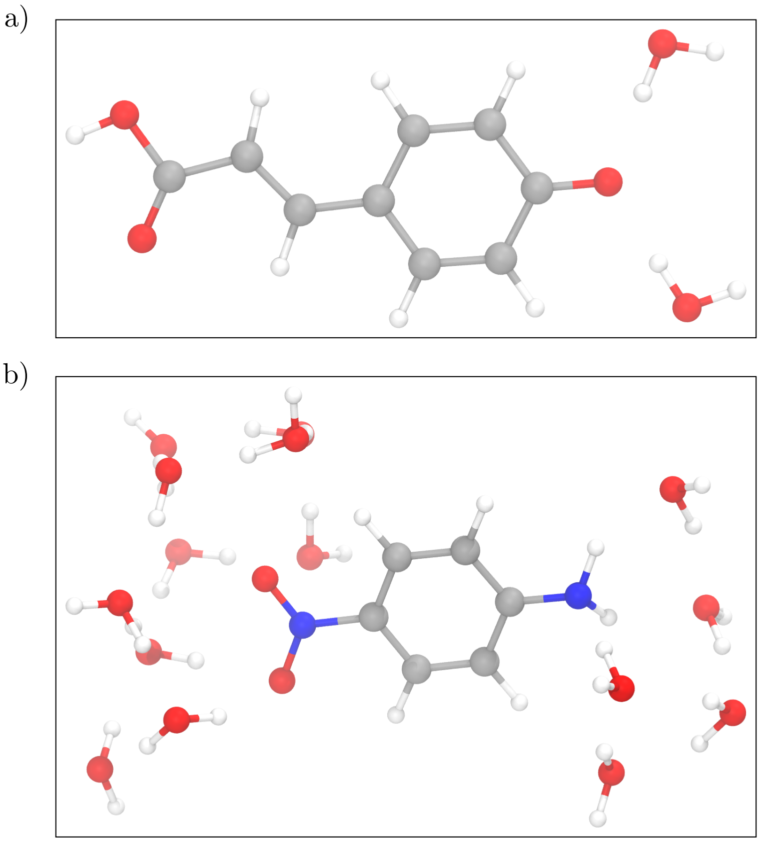


Figure 5.2: Molecular structures of the test systems a) PYPb/2 H_2O and b) *p*NA/15 H_2O. Water molecules comprise the polarizable environment, whereas the solute molecules are used as the quantum region.

entire *p*NA/15 H_2O supersystem as benchmark reference values. For the energetically lowest singlet $\pi \rightarrow \pi^*$ transition, the two-photon transition strength was computed. These quantities were also obtained using PE-ADC(2)/6-31+G* with PYPb as quantum region and the two water molecules as polarizable environment. Different PE coupling schemes were used, i.e., PTE-PE and LR-PE, combined with PE(ECP) repulsive potentials and EEF for transition moments, yielding six different combinations of coupling scheme and additional correction (PTE-PE, PTE-PE + ECP, LR-PE, LR-PE + EEF, LR-PE + ECP, LR-PE + ECP + EEF). From a conceptual point of view, it only makes sense to use EEF in combination with the LR-PE formalism.[122] These computations were carried out using ADC(3/2), too. The results of the ADC(2) calculations for PYPb/2 H_2O are presented in Table 5.1. Note that excitation energies are of course not affected by EEF. For such a small system with only two water molecules as polarizable environment, the excitation energy errors are almost negligible, as expected. The largest error is found for the simplest

Table 5.1: Excitation energies E_{exc}, oscillator strengths f, and TP transition strengths δ_{TP} for the low-lying $\pi \rightarrow \pi^*$ state of PYPb/2 H_2O obtained with ADC(2)/6-31+G* using different environment coupling schemes.[a)]

Scheme	E_{exc} [au]	f	δ_{TP} [au]
Supersystem	0.109	0.902	32494.877
PTE-PE	0.113 (2.84)	0.874 (−3.09)	27576.994 (−15.13)
PTE-PE + ECP	0.109 (−0.37)	0.877 (−2.75)	27727.543 (−14.67)
LR-PE	0.111 (1.87)	0.883 (−2.12)	28386.9 (−12.64)
LR-PE + EEF	0.111 (1.87)	0.918 (1.73)	30808.337 (−5.19)
LR-PE + ECP	0.108 (−1.35)	0.885 (−1.86)	28530.88 (−12.2)
LR-PE + ECP + EEF	0.108 (−1.35)	0.919 (1.86)	30871.888 (−4.99)

[a)] Relative errors in % with respect to the supersystem reference are shown in parentheses.

PTE-PE scheme, however, all relative errors are smaller than 3%. Regarding oscillator strengths, the LR-PE scheme improves upon PTE-PE by approximately one percentage point. As expected, the most accurate oscillator strengths are obtained when using EEF to compute transition moments. This observation is even more pronounced for TP transition strengths. Here, the largest discrepancy is found for PTE-PE, i.e., again the most simple scheme. Including the PE coupling term in the ADC matrix for the eigenvalue problem and linear response procedure reduces the error by three percentage points, however, the δ_{TP} value is still underestimated by approximately 13%. The most accurate δ_{TP} value is obtained with the LR-PE + ECP + EEF combination, i.e., with LR coupling and both corrections. The effect of PE(ECP) repulsive potentials is, however, negligible in this case, since a similar error of approximately 5% results from LR-PE with just EEF. The expected improvement through LR-PE in combination with finite field effects was found in this first assessment. Turning to the ADC(3/2) results, shown in Table 5.2, a similar trend is observed. All relative excitation energy errors are below 3% in magnitude. Interestingly, the relative errors of δ_{TP} are much smaller for ADC(3/2) than for ADC(2). The most accurate result is here found for the combination of LR-PE and EEF. Adding PE(ECP) to this setting slightly increases the relative error. A key result from this analysis is that in any case EEF should be used to obtain more precise transition intensities with the LR-PE model. To the best of my knowledge, this has not been done before in the previously existing PE-ADC implementation. Furthermore, this is the first LR-PE-ADC(3/2) computation conducted, combining a highly accurate electronic structure method in a robust manner with the PE model. To assess the performance of the methodological combinations for a

Table 5.2: Excitation energies E_{exc}, oscillator strengths f, and TP transition strengths δ_{TP} for the low-lying $\pi \rightarrow \pi^*$ state of PYPb/2 H_2O obtained with ADC(3/2)/6-31+G* using different environment coupling schemes.[a)]

Scheme	E_{exc} [au]	f	δ_{TP} [au]
Supersystem	0.115	0.886	14669.096
PTE-PE	0.117 (1.79)	0.843 (−4.92)	13552.182 (−7.61)
PTE-PE + ECP	0.113 (−2.17)	0.835 (−5.8)	14036.726 (−4.31)
LR-PE	0.116 (1.09)	0.852 (−3.92)	13846.046 (−5.61)
LR-PE + EEF	0.116 (1.09)	0.883 (−0.4)	14857.739 (1.29)
LR-PE + ECP	0.112 (−2.87)	0.843 (−4.88)	14340.235 (−2.24)
LR-PE + ECP + EEF	0.112 (−2.87)	0.873 (−1.52)	15357.64 (4.69)

[a)] Relative errors in % with respect to the supersystem reference are shown in parentheses.

larger system, an additional test case is presented. The composite structure of pNA with 15 water molecules (pNA/15 H_2O) was optimized at the CAM-B3LYP/cc-pVDZ[127,128] level of theory using TeraChem 1.93.[129] The resulting molecular geometry is shown in Figure 5.2b. The embedding parameters were obtained exactly as described above for the PYPb system. Three excited singlet states were obtained at the ADC(2)/6-31+G* level of theory for the entire pNA/15 H_2O supersystem as the benchmark reference. For the energetically lowest singlet $\pi \rightarrow \pi^*$ transition, the two-photon transition strength δ_{TP} was computed. These quantities were also obtained using PE-ADC(2)/6-31+G* with pNA as quantum region and the water molecules as polarizable environment. The same method combinations were used as described above for the PYPb/2 H_2O system above. Since the supersystem is quite large, an ADC(3/2) is unfortunately not feasible. The results for the pNA/15 H_2O system are shown in Table 5.3. Because the environment is now comprised of 15 water molecules, the overall relative errors are naturally somewhat larger than in the previous test case. Still, the deviation of excitation energies of the low-lying $\pi \rightarrow \pi^*$ singlet state from the supersystem ADC(2) reference are all smaller than 7%. The most accurate excitation energy is obtained for LR-PE in combination with PE(ECP), showing a deviation as small as 2.35%. For the oscillator strength f, the error of PTE-PE of approximately −13% can be reduced to the remarkable small deviation of 1.8% with LR-PE + ECP + EFF. This error reduction is even more pronounced for the higher-order property δ_{TP}, where the error reduced from −44.45% in PTE-PE to only −9.4% when using the most advanced LR-PE + ECP + EEF scheme. One can clearly see that the combination of the PE linear response contribution with EEF and PE(ECP) is beneficial

Table 5.3: Excitation energies E_{exc}, oscillator strengths f, and TP transition strengths δ_{TP} for the low-lying $\pi \rightarrow \pi^*$ state of pNA/15 H_2O obtained with ADC(2)/6-31+G* using different environment coupling schemes.[a)]

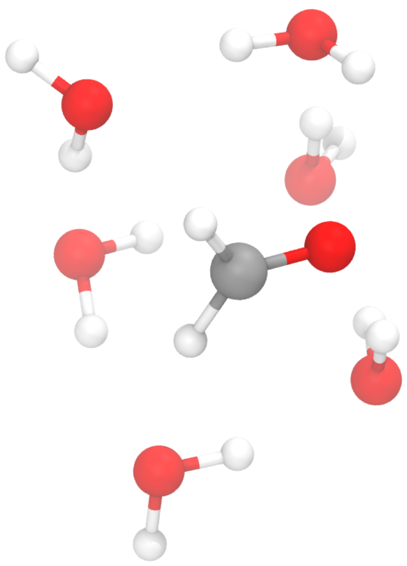
Scheme	E_{exc} [au]	f	δ_{TP} [au]
Supersystem	0.147	0.500	9970.274
PTE-PE	0.157 (6.67)	0.434 (−13.24)	5538.653 (−44.45)
PTE-PE + ECP	0.153 (3.89)	0.431 (−13.85)	6575.304 (−34.05)
LR-PE	0.155 (5.22)	0.457 (−8.63)	5901.168 (−40.81)
LR-PE + EEF	0.155 (5.22)	0.514 (2.72)	7557.326 (−24.2)
LR-PE + ECP	0.151 (2.35)	0.446 (−10.73)	6941.542 (−30.38)
LR-PE + ECP + EEF	0.151 (2.35)	0.509 (1.8)	9033.5 (−9.4)

[a)] Relative errors in % with respect to the supersystem reference are shown in parentheses.

especially for higher-order properties. Relying only on the approximate PTE scheme, as is done frequently for similar models,[70] yields much larger errors. Even though only two test cases are discussed here, it is likely that the combined LR-PE-ADC methodology performs well for molecular response properties of higher order when (modified) transition moments are computed with EEF, and electron spill-out artifacts are avoided through PE(ECP). For electronic excitation energies only, however, little improvement was observed in the presented benchmarks, where one could also just use perturbative corrections, for example. It is evident that the PTE error largely amplifies with increasing order of the molecular property under study, such that an LR-PE-ADC scheme in combination with EEF and PE(ECP) should preferably be used in this case. The synergy of this first-time combined methodology looks promising for accurate description of nonlinear response properties in, e.g., photoreactive proteins.[130]

5.3 Analytic Nuclear Gradients

To verify the correctness of the entire PE gradient suite, especially PE-MP and PE-ADC gradients derived in Chapter 3, I carried out a small set of test calculations. The test system is a formaldehyde (FA) molecule as the quantum region, microsolvated by six water molecules. This structure was taken from Ref. 64. Static charges and dipole moments as well as dipole-dipole polarizabilities were obtained using LoProp[124] through PyFraME[125] at the PBE0/loprop-cc-pVDZ level of theory as implemented in Dalton.[88,126,128,131] The test system structure is illustrated in Figure 5.3. All calculations were carried out us-



	x	y	z
C	1.063	2.027	0.434
O	1.115	1.08	1.154
H	1.094	3.039	0.836
H	0.984	1.924	−0.645
O	1.379	−1.352	−0.484
H	1.186	−0.722	0.214
H	2.294	−1.592	−0.324
O	1.425	0.333	−2.729
H	1.307	−0.349	−2.064
H	1.134	−0.089	−3.54
O	4.294	−1.306	0.061
H	4.956	−1.963	0.286
H	4.292	−0.718	0.819
O	3.982	2.785	0.137
H	4.094	2.228	−0.636
H	4.672	3.444	0.043
O	4.168	0.934	−1.933
H	3.321	0.8	−2.364
H	4.323	0.104	−1.477
O	4.048	0.636	2.075
H	3.147	0.623	2.406
H	4.086	1.444	1.559

Figure 5.3: Graphical illustration of the FA test system with six water molecules as polarizable environments (left) and Cartesian coordinates in Å (right). The atom labels marked in red belong to FA, i.e., the quantum region.

ing PySCF as host program interfaced with CPPE,[39,79] using the cc-pVDZ basis set.[128] Numerical nuclear gradient values were obtained using a five-point finite difference scheme with a step size of 10^{-3} au, and agreement with analytical values was tested with a tolerance of 10^{-8} au. First, the simplest gradient implementation in PySCF/CPPE was tested by computing the analytical PE-HF gradient for the test system structure. These results, together with numerical reference values are shown in Table 5.4. Here, the analytical PE-HF gradient values perfectly agree with the numerical result obtained from finite differences, proving that the implementation and all gradient building blocks are indeed correct. Note that my implementation supports *all* SCF ground state methods implemented in PySCF, i.e., PE-DFT gradients are enabled, too. Furthermore, I implemented PE-SCF gradients in Psi4, and the Psi4 implementation was found to be correct by comparison with finite difference values (data not shown). With the correct PE-HF gradient implementation in place, the next step was to carry out a correlated ground state PE-MP2 gradient computation within the PTE scheme. For this task, PySCF was again used as host program for `adcc`, and the implementation of all correlated gradients was achieved as described in Section 4.2.4. As a prerequisite for working PE gradients, I tested the vacuum MP and ADC gradients with the same numerical scheme, omitted here for brevity. The

Table 5.4: PE-HF/cc-pVDZ gradient components for the FA molecule, microsolvated with six water molecules.[a)]

	Analytical			Numerical		
	$\mathrm{d}E/\mathrm{d}x$	$\mathrm{d}E/\mathrm{d}y$	$\mathrm{d}E/\mathrm{d}z$	$\mathrm{d}E/\mathrm{d}x$	$\mathrm{d}E/\mathrm{d}y$	$\mathrm{d}E/\mathrm{d}z$
C	−0.0021497	0.0066146	−0.0105669	−0.0021497	0.0066146	−0.0105669
O	−0.0032942	0.0017999	0.0043360	−0.0032942	0.0017999	0.0043360
H	0.0000844	−0.0062130	−0.0003124	0.0000844	−0.0062130	−0.0003124
H	0.0002680	0.0002660	0.0090324	0.0002680	0.0002660	0.0090324

a) All values in atomic units.

PTE-PE-MP2 gradient components for the FA test system are presented in Table 5.5. Clearly, the analytically and numerically obtained values for all gradient components are equivalent up to a threshold value of 10^{-8} au again. This shows that the derivation for all orbital response contributions as well as the gradient contributions through the correlation treatment do not contain any errors. The rate-limiting step for the final evaluation of the correlated gradient is the `GradientProvider` implementation in which the contraction of the densities with derivative integrals is carried out. Therein, one could possibly play some tricks with PySCF to avoid evaluation of zero or redundant gradient contributions, which would make the implementation more usable for larger systems. This should be easily achievable, because the PE-MP2 implementation is working properly and can be reliably tested. For further testing, I planned to run the exact same computation using the PE-MP2 gradient implementation in Turbomole 7.5,[32,132] however, the feature was unfortunately disabled. The last and most intricate feature of the presented PE suite of gradient implementations is the evaluation of the LR-PE-ADC excited state gradient.

Table 5.5: PTE-PE-MP2/cc-pVDZ gradient components for the FA molecule, microsolvated with six water molecules.[a)]

	Analytical			Numerical		
	$\mathrm{d}E/\mathrm{d}x$	$\mathrm{d}E/\mathrm{d}y$	$\mathrm{d}E/\mathrm{d}z$	$\mathrm{d}E/\mathrm{d}x$	$\mathrm{d}E/\mathrm{d}y$	$\mathrm{d}E/\mathrm{d}z$
C	0.0003620	−0.0274673	0.0161553	0.0003620	−0.0274673	0.0161553
O	−0.0053668	0.0466184	−0.0305970	−0.0053668	0.0466184	−0.0305970
H	−0.0000706	−0.0166210	−0.0025885	−0.0000706	−0.0166210	−0.0025885
H	0.0010418	−0.0006693	0.0191839	0.0010418	−0.0006693	0.0191839

a) All values in atomic units.

To obtain well-converged eigenvectors, the five energetically lowest singlet excited states for the microsolvate FA system were computed with LR-PE-ADC(2)/cc-pVDZ as implemented in `adcc` using PySCF and CPPE. The convergence tolerance of the eigensolver was set to 10^{-8} au. Then, both numerical and analytical gradients were evaluated for the energetically lowest singlet state ($E_{\mathrm{exc}} = 0.15960386$ au), summarized in Table 5.6. Even though the LR-PE-ADC gradient expressions are somewhat involved, the result obtained with my analytical implementation is in perfect agreement with the numerical result. This proves, without a doubt, the correctness of all the working equations and the subsequent implementation in `adcc`. As an additional test calculation, I tried again to run the exact same LR-PE-ADC(2) gradient computation using Turbomole 7.5, providing the exact same input parameters, molecular geometry, and basis set. Since the Turbomole implementation uses the resolution of the identity (RI) approximation, I selected the aug-cc-pV6Z auxiliary basis set[133] to minimize the error of the RI approximation. The PE-HF and PTE-PE-MP2 energy as well as the LR-PE-ADC(2) excitation energy were in exact agreement with my results from `adcc` up to numerical accuracy. To my surprise, an erroneous LR-PE-ADC(2) gradient was returned by Turbomole, shown in Table 5.7. The equality of all energy terms clearly rules out a possible problem with the preparation of the calculation. Hence, if the energy itself agrees among implementations, then also its derivative must be identical. In addition, the numerical gradient values largely differ from the Turbomole result, such that a numerical problem can be excluded. These errors are on the order of 10^{-4} au, definitely too large considering the agreement among ground and excited state energies. Consequently, there exists a problem with the current LR-PE-ADC(2) gradient implementation in Turbomole, which can unfortunately not be tracked down for such a closed-source code.

Table 5.6: LR-PE-ADC(2)/cc-pVDZ gradient components for microsolvated FA.[a)]

	Analytical			Numerical		
	$\mathrm{d}E/\mathrm{d}x$	$\mathrm{d}E/\mathrm{d}y$	$\mathrm{d}E/\mathrm{d}z$	$\mathrm{d}E/\mathrm{d}x$	$\mathrm{d}E/\mathrm{d}y$	$\mathrm{d}E/\mathrm{d}z$
C	0.0115224	−0.1511810	0.1106262	0.0115224	−0.1511810	0.1106262
O	−0.0110975	0.1619811	−0.1213490	−0.0110975	0.1619811	−0.1213490
H	−0.0019642	−0.0123892	−0.0044773	−0.0019642	−0.0123891	−0.0044773
H	0.0002861	0.0023864	0.0167721	0.0002861	0.0023864	0.0167720

[a)] All values in atomic units.

The sample computations presented above demonstrated the correctness of the PE gradient derivations and implementations carried out in this work. All combined methodolo-

Table 5.7: LR-PE-ADC(2)/cc-pVDZ/RI-aug-cc-pV6Z gradient components for FA obtained with Turbomole 7.5.[a)]

	$\mathrm{d}E/\mathrm{d}x$	$\mathrm{d}E/\mathrm{d}y$	$\mathrm{d}E/\mathrm{d}z$
C	0.01141519	−0.1502619	0.1102927
O	−0.01103455	0.1590101	−0.1185370
H	−0.00192592	−0.0103326	−0.0038015
H	0.00011647	0.0021209	0.0141014

a) All values in atomic units.

gies, for which PE gradient were implemented, perfectly agreed with numerical nuclear gradient results. PE gradients for SCF reference states are available via the PySCF and Psi4 host programs, whereas PE-MP2 and PE-ADC gradients can be computed using `adcc` with either backend.

5.4 Summary

In this chapter, I presented possible applications of the combined PE-ADC methodologies. For example, the perturbative approach relying on corrections for excitation energies was successfully tested in supersystem computations and in comparison to continuum solvation models. The most advanced application of the pt-PE-ADC method so far is the investigation of the CT state in the photocycle of the dodecin protein.[61] Using my combined method, I showed that the polarizable protein environment directly promotes this key step for efficient protection of embedded flavin species. Moreover, a first set of benchmark computations for the novel LR-PE-ADC method applied to molecular response properties was shown. Employing this more advanced coupling scheme together with corrected transition intensities through EEF and removal of electron spill-out artifacts with PE(ECP), highly accurate two-photon transition strengths were obtained for small water-solvated chromophores. The correctness of PE gradients for ground and excited states was verified by comparison to numerical schemes. I want to stress that the entire feature set presented in this thesis is freely available, as explained in Chapter 4. Almost every ADC-related feature is already implemented in the most recent `adcc` release, the rest is available through feature branches. The unique and comprehensive feature set of combined PE-ADC methodologies for excited states, molecular response properties of any order, and analytic gradients cannot be found in any other quantum chemical program package. In addition, this chapter showed the first combination of the PE model with

ADC/ISR for response properties, taking into account EEF and PE(ECP). Even though only few applications are shown herein, the methodologies are promising for future research because they are easily extensible, easy to used, and new features coming to `adcc` will work with PE out of the box. Currently, the PE-ADC gradient code is difficult to use in practice due to performance bottlenecks. Improving the speed of the implementation and ad hoc addition of an empirical dispersion interaction [59] between the environment and the quantum region will enable efficient PE-ADC geometry optimizations in the future. To assess the quality of PE-ADC results in general, however, an in-depth analysis of the performance on several systems will be required. This will pave the way for application of, e.g., LR-PE-ADC to non-linear spectroscopies of biomolecular systems. [130]

Notes

1. Reprinted with permission from: M. Scheurer, M. F. Herbst, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted, "Polarizable Embedding Combined with the Algebraic Diagrammatic Construction: Tackling Excited States in Biomolecular Systems", *J. Chem. Theory Comput.* **2018**, *14 (9)*, 4870-4883. Copyright 2018 American Chemical Society.

Chapter 6

Linear-Scaling Polarizable Embedding

For polarizable embedding models, a large share of the computational cost arises from evaluation of the interaction integrals for the quantum-classical coupling, i.e., multipole potential and field integrals. While for small environments, these contributions surely dominate the additional computational cost, calculations with ever larger environments are dominated by the contributions with the steepest asymptotic scaling. All PE models require the solution of classical polarization equations for the induced dipole moments, which scales quadratically with the number of polarizable sites in the environment. Spurred by this bottleneck, linear-scaling implementations relying on the Fast Multipole Method (FMM)[134] have been developed in recent years.[135,136] The scaling of iteratively evaluated terms arising in PE models is illustrated in Figure 6.1. The quantum→classical polarization consists of density-dependent electric field expectation values $\boldsymbol{\mathcal{F}}_{\text{el}}[\mathbf{D}]$ evaluated at each polarizable site in the environment, as previously explained. The classical→quantum polarization is, in addition to the presence of the permanent multipoles in the environment, described by the induction operator $\hat{V}^{\text{ind}}$ taking into account the induced field at each environment site. Hence, both terms scale linearly with the number of polarizable sites. The key step in the PE model is to solve the linear equations for induced dipole moments $\boldsymbol{\mu}_{\text{ind}}$ (eq (2.143)). The induced fields $\boldsymbol{\mathcal{F}}_{\text{ind}}$ in eq (2.142) are evaluated for each pair of polarizable sites, such that this term scales quadratically with the number of sites. Using FMM, this bottleneck can be mitigated, and as a result $\boldsymbol{\mathcal{F}}_{\text{ind}}$ scales linearly as well. Similarly, the evaluation of the static multipole field $\boldsymbol{\mathcal{F}}_{\text{mul}}$ in eq (2.140) scales quadratically with the number of sites, however, it must only be evaluated once during a computation

---

Parts of this chapter have already been published in:

- M. Scheurer, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted, “Efficient Open-Source Implementations of Linear-Scaling Polarizable Embedding: Use Octrees to Save the Trees”, *J. Chem. Theory Comput.* **2021**, *17 (6)*, 3445–3454.

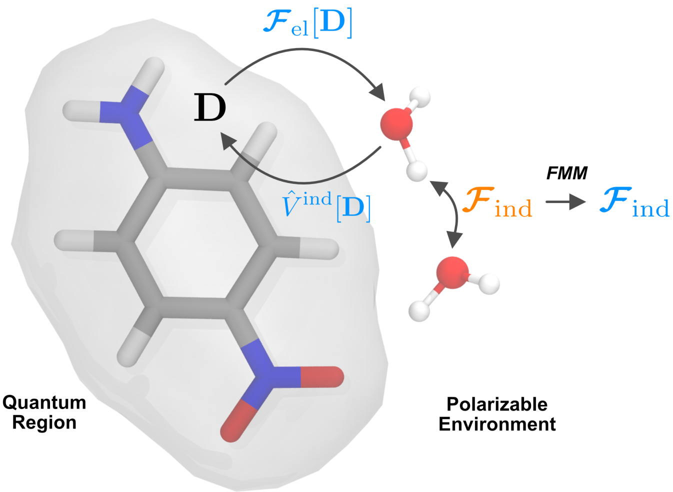


Figure 6.1: Illustration of iteratively evaluated interaction terms in PE models. Terms that scale linearly with the number of polarizable sites N_{sites} are shown in blue, whereas quadratically scaling terms are shown in orange. Using FMM, the quadratically scaling term to compute the induced fields $\boldsymbol{\mathcal{F}}_{\mathrm{ind}}$ at all polarizable sites becomes linearly scaling.[1]

and can then be cached for further use. Very detailed derivations and analyses of FMM can be found in the literature,[137–139] but in brief, FMM consists of the following key steps: The system (sites in the environment) is subdivided in hierarchical boxes using an octree. An octree is a three-dimensional spatial data structure with cubic boxes at different levels, i.e., the boxes can have a “parent” box and, if they are not leaf nodes, eight child boxes. The boxes are sometimes referred to as nodes as well. After the recursive tree creation, multipole expansions of each node are computed through a given expansion order p. These multipole expansions are started at the leaf level of the octree by a mapping from source particles to the multipole expansion (particle to multipole, P2M). Then, the expansions are translated to higher-level tree nodes up to the root nodes (multipole to multipole, M2M). Based on a so-called angle opening criterion, with the opening angle θ, interactions lists of the tree nodes are generated. These are required to bookkeep which cell-cell interactions are directly evaluated (particle to particle, P2P), i.e., via direct summation, or via their far-field expansions. Interactions are evaluated using multipole expansions if the cells are well separated, i.e., if they satisfy $R\theta < R_A + R_B$, where R is the distance between the cell centers, and R_A and R_B are the sizes of cells A and B. Then, short-ranged interactions (P2P) are directly evaluated, if the cells are too close to interact via multipole expansions. The potential/fields of far-field nodes are computed (multipole to local, M2L) and transferred down the octree to the leaf level (local to local, L2L). Finally, the long ranged part of the potential/field is evaluated by means of the local expansion (local to particle, L2P). With this approach, the potential/field can then be evaluated at a cost that scales asymptotically as $\mathcal{O}(N_{\text{sites}})$. While the above outline is rather simplistic, an actual FMM implementation requires caution to ensure high and tunable performance and accuracy.

The quadratic scaling of the induced field equations becomes particularly problematic for large environments when a lot of induced moment equations need to be solved. This is the case for computation of molecular response properties within the PE framework, especially for high-order response properties, where the induction operator must be obtained for many trial vectors. Compared to the iterations needed to converge the PE-SCF electronic ground state, the number of polarization equations that need to be solved to obtain, e.g., excitation energies and transition moments is much larger.

All previously presented FMM implementation in the context of PE models are found in closed-source codes and are not publicly available. To this end, I teamed up with P. Reinholdt to develop a linear-scaling formulation of the PE model. We could achieve two independent FMM implementations in two open-source libraries, PElib[87] and CPPE.[79,140] The implementation in PElib by P. Reinholdt exposes PE-FMM to the Dalton program package,[88] and interfaces to LSDalton[88] and DIRAC[141] will be available soon. Details and benchmarks of this implementation can be found in the joint publication.[140] In the following, I will only discuss the implementation and results of my implementation in CPPE, which currently enables the linear-scaling functionalities in PySCF and Psi4.[38,39] After presenting the implementation and computational details, benchmark analyses are carried out, and the implementation is tested using real-world sizable biomolecular systems with more than a million polarizable sites in the environment.

6.1 Computational Methodology

6.1.1 Implementation Details

The implementation in the CPPE library[79] is based on autogenerated C++ code using fmmgen.[139] I followed the design strategy for CPPE outlined in Chapter 4, that is, use existing frameworks to achieve a single task. Some modifications to the fmmgen code generator were required, as explained below. Still, the implementation with reproducible, automatically generated code is sustainable and was finished with minor time and programming effort. If at some point another FMM code should be interfaced with CPPE, this is straightforward because only the function calls to the evaluation of electric fields need to be refactored. The fmmgen library generates Cartesian operators and all kernels required for FMM based on symbolic algebra with SymPy.[102] The script I used to generate the FMM code for CPPE can be found in the CPPE GitHub repository (`https://github.com/maxscheurer/cppe`). I enabled common subexpression elimination (CSE), as for the T-tensor code generator, and let the code generator expand the power function `std::pow(b, n)` as products up to $n = 11$. To allow for damped P2P kernels and streamlined code for multiple source orders, the fmmgen code was slightly adapted (`https://github.com/fmmgen/maxscheurer`). For the latter, I used template meta-programming such that the code path for each source order is fixed at compile time. This means that the individual kernels posses two template arguments, one being the order of the source multipole moment `M`, the second one is the output vector size `osize`. For example, the code to compute the direct electric field interaction of a charge is accessible via the function `P2P<0, 3>(...)`. With this strategy, the computation code is rather concise as it is not polluted with `if`-statements to select the appropriate code path. The generated code comes with loop-based OpenMP parallelism included. The tree code in CPPE was adapted from the fmmgen code examples and interfaced to the low-level routines which require electric field computations. Of course, host programs that interface with CPPE do not require any changes to adapt for the linear-scaling FMM code except for exposing the additional options (which is a trivial change). FMM is available in CPPE as of version 0.3.1.

6.1.2 Test System Setup

The implementation was tested on the pNA molecule placed inside a water solvation box. The pNA geometry was taken from Ref. 64 and solvated using the PACKMOL package.[142] Eleven systems with different box sizes were generated, ranging from 1440 atoms to 193596 atoms. The system with 114465 polarizable sites (38155 H_2O molecules) will be referred to as pNA/38k H_2O in the following, as it is used in several test cases. The generated systems were not processed further because PACKMOL generates configurations without steric clashes, such that they can directly be used for benchmark purposes. The environment water molecules were parametrized using LoProp[124,126] at the CAM-B3LYP/6-31+G* level of theory.[127,143] The parametrization of a single water molecule was run with PyFraME[125] and Dalton,[88] and parameters for the remaining waters were then assigned by translation and rotation of the parameters from the reference molecule. Thus, each atom is assigned a charge, dipole moment, quadrupole moment, and anisotropic dipole–dipole polarizability, i.e., all atoms in the test systems are polarizable.

These benchmark systems were used for the PElib implementation in the joint publication, too.[140]

6.1.3 Benchmark and Test Calculations

To assess the correctness and accuracy of the FMM implementation, initial testing was performed on the systems described above. A parameter study for the expansion order p and the opening angle θ was run by evaluating the static electric field $\boldsymbol{\mathcal{F}}_{\text{mul}}$ of the pNA/38k H_2O test system at all sites. The field evaluations from FMM were then compared to results from direct summation. Serial timings of the field evaluations were recorded to assess the performance for different FMM parameters. These calculations were run on an Intel Xeon E5-2680 v3 processor using the Intel 19 compiler. The choice of compiler can have quite a large impact on the performance of the fmmgen-generated code, and it was shown in the fmmgen publication that Intel compilers outperform, e.g., GCC in this case.[139] Next, the linear equations to solve the induced dipole moments (eq (2.143)) were benchmarked for all water boxes generated in Section 6.1.2 with the field from static multipole moments $\boldsymbol{\mathcal{F}}_{\text{mul}}$ as the right-hand side of the equation, i.e., $\mathbf{B}\boldsymbol{\mu}_{\text{ind}} = \boldsymbol{\mathcal{F}}_{\text{mul}}$. No parallelization was used in these computations. The equations were solved using a Jacobi-preconditioned Conjugate Gradient (PCG) algorithm with a residual norm convergence threshold of 10^{-8} au. The same convergence threshold was used for all other calculations presented in this chapter.

The linear scaling in production PE-SCF and PE-TDDFT calculations, where a multitude of field evaluations is required, was then tested on all generated pNA systems. Calculations were run with PySCF[39] at the TD-CAM-B3LYP[127]/6-31G(d,p) level of theory using the Tamm-Dancoff approximation (TDA).[144] The five energetically lowest singlet excited states were computed. All of these computations were run on a single node with 24 OMP threads. Timings for the SCF and the linear response (LR) procedure were recorded separately. All calculations were run employing i) direct summation and ii) FMM with $p = 5$ and $\theta = 0.5$.

The five lowest singlet excitation energies of pNA/38k H_2O were computed with pt-PE-ADC(2) and the 6-31G(d,p) basis set. PE-ADC(2) calculations were run with PySCF[39] in the adcc[80] toolkit employing ptSS and ptLR corrections for the excitation energies, i.e., the induced moment equations are not solved during the ADC procedure itself.[61] These computations were run employing either i) direct summation and ii) FMM with $p = 5$ and $\theta = 0.5$.

Table 6.1: Nile Red/BLG Systems.

System Name	Unit Cells	N_{sites}
1x1x1	1	61,634
3x1x1	3	184,902
3x3x1	9	554,706
3x3x3	27	1,664,118

6.1.4 Setup for Nile Red

A snapshot extracted from a QM/MM MD trajectory of the Nile Red molecule bound to a Beta-lactoglobulin protein (BLG) was taken from a previous study.[22] The coordinates of the Nile Red molecule were extracted directly and used as the QM region in the following. Embedding parameters of the protein, water, and ions were assigned using the PyFraME package. For the protein, the parameters described in Ref. 145 were used, while parameters for the water molecules and ions were taken from Ref. 146. In addition to the primary simulation cell, we also created larger 3x1x1, 3x3x1 and 3x3x3 replicated cells with 3, 9, and 27 copies of the primary unit cell, respectively, which for the largest system yielded a total of 1,664,118 environment sites, as summarized in Table 6.1. For the PE calculations, FMM was used with the parameters set to $p = 5$ and $\theta = 0.5$. Thole-style damping was used for all systems using standard damping parameters.[103] Three singlet excited states were computed using PE-ADC(2)/6-31G(d) with CPPE/PySCF in adcc[80] including ptSS and ptLR corrections (see Section 3.1).[61] The ADC calculations were run on a single node using 24 OpenMP threads. The resulting stick spectra were convoluted with a Lorentzian broadening function with a half-width at half maximum value of 0.124 eV. The code used to generate the pNA test systems and to produce the following plots can be found on GitHub (`https://github.com/maxscheurer/pe_fmm`).

6.2 Results and Discussion

6.2.1 Errors and Timings of Electric Field Evaluations

To test the accuracy of electric field evaluations of the implementation, a parameter study varying the expansion order p and the opening angle θ for the pNA/38k H_2O test system was conducted.[140] Static multipole fields $\mathcal{F}_{\text{mul}}$ with contributions from static charges, dipoles, and quadrupoles were computed. The error of the static multipole field per site i

is given by

$$\mathcal{F}_i^{\mathrm{err}} = \frac{\left\| \boldsymbol{\mathcal{F}}_{\mathrm{mul},i}^{\mathrm{direct}} - \boldsymbol{\mathcal{F}}_{\mathrm{mul},i}^{\mathrm{FMM}} \right\|}{\left\| \boldsymbol{\mathcal{F}}_{\mathrm{mul},i}^{\mathrm{direct}} \right\|}. \tag{6.1}$$

The normalized error distributions for $\theta = 0.2, 0.3, 0.5, 0.7, 0.99$ and $p = 3, 5, 7$ are shown in Figure 6.2. For small opening angles and high expansion orders, the errors in the field evaluation are almost negligible due to the fact that a lot of field evaluations are identical to direct summation. As a matter of fact, an opening angle close to 1.0 should not be used in practice as the errors simply become too large. It is clear that the average error can be easily controlled using both the opening angle and the expansion order in a well-defined manner, comparable to previous implementations.[135] Since induced fields are evaluated iteratively during the SCF or LR procedures, the overall errors due to FMM field evaluations will, of course, accumulate. This will be analyzed in the following. The presented error distributions show that the accuracy of the implementation are well controllable and approach the result from direct summation for large expansion orders and/or small

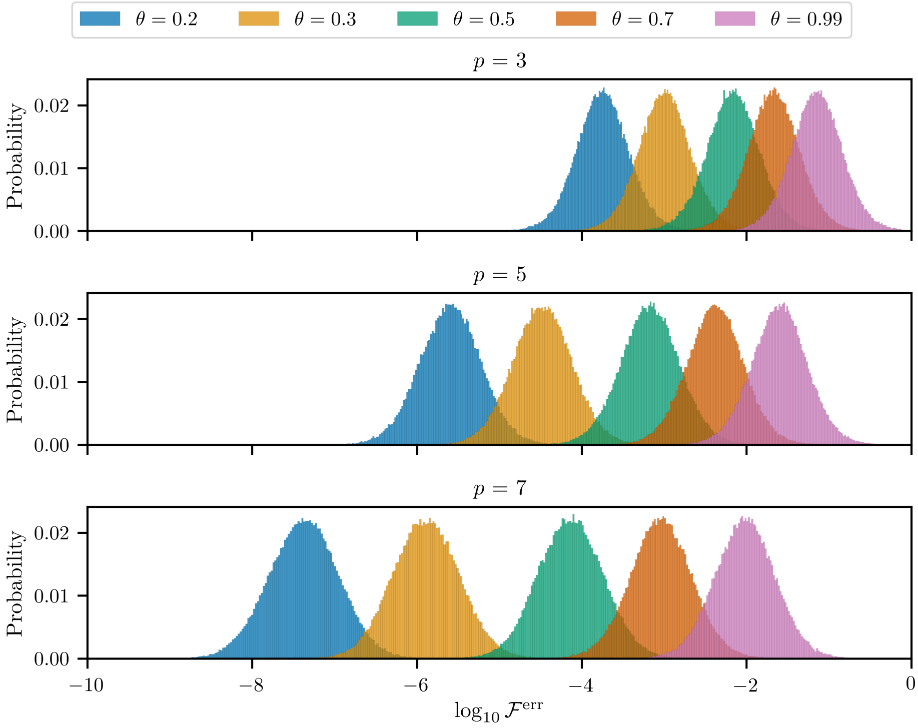


Figure 6.2: Accuracy of FMM electric fields. The histograms show probability densities over the error defined in eq (6.1). The errors are shown for different values of the expansion order p and opening angle θ for a static multipole field $\boldsymbol{\mathcal{F}}_{\mathrm{mul}}$ evaluation in pNA/38k H_2O. Data reproduced from Ref. 140.

opening angles. The results found for the PElib implementation are comparable to CPPE, which gave further assurance.[140]

Another aspect of ensuring the correctness of the implementation is the time spent on the field evaluations. The trade-off between computational effort and accuracy is a significant factor in the choice of p and θ for practical simulations. The timings for the field evaluations in Figure 6.2 are summarized in Table 6.2, obtained without using any parallelization scheme. The run time of the direct summation evaluation is about 1500 seconds. The execution time is reduced by a factor of two with the most conservative (and slow) FMM parameters tested, whereas the fastest parameter set is up to 130–260 times faster. The run time increases with increasing expansion orders and decreasing opening angle, as expected. To put the timings in perspective with respect to the entailed errors, we observe that for $(p = 5, \theta = 0.50)$, the evaluation time decreases by approximately a factor of 20–25, while the average $\mathcal{F}_i^{\mathrm{err}}$ is only 10^{-3}. Our results qualitatively agree with a similar accuracy analysis conducted by Caprasecca et al.[147] In summary, the presented errors and timings show the anticipated behavior and allowed us to efficiently evaluate electric fields with the expected error-performance trade-off. The performance of the PElib FMM implementation was rather similar.[140] Based on both the timings and the field evaluation accuracy, the combination of $(p = 5, \theta = 0.50)$ was selected as default in the following analyses, since it gives a reasonable compromise between speed and errors.

Table 6.2: Serial timings for static field evaluations $\boldsymbol{\mathcal{F}}_{\mathrm{mul}}$ in Figure 6.2[a), b)]

θ	$p = 3$	$p = 5$	$p = 7$
0.20	668.20	666.66	692.21
0.30	255.36	312.30	272.18
0.50	69.00	76.86	74.62
0.70	33.26	34.01	35.99
0.99	12.17	12.36	13.52
Direct		1574.63	

a) Timings are reported in seconds.
b) Data reproduced from Ref. 140.

Next, the performance of the fmmgen-based implementation was benchmarked by solving the linear equations for the induced dipole moments. The total time to solve the equation $\mathbf{B}\boldsymbol{\mu}_{\mathrm{ind}} = \boldsymbol{\mathcal{F}}_{\mathrm{mul}}$, number of iterations, and time per iteration are shown in Table 6.3. Since the induced fields $\boldsymbol{\mathcal{F}}_{\mathrm{ind}}$ are evaluated for the matrix-vector product in each iteration, the timings for FMM clearly shows the expected performance improvement: For

Table 6.3: Timings for solution of the induced dipole moments with direct summation and FMM.[a), b)]

	Time [s]		N_{iter}		Time / N_{iter} [s]	
N_{sites}	Direct	FMM	Direct	FMM	Direct	FMM
1440	0.9	0.8	10	10	0.1	0.1
4083	7.2	3.6	10	10	0.7	0.4
8805	37.6	9.4	11	11	3.4	0.9
16203	127.0	29.4	11	11	11.5	2.7
26883	347.5	56.4	11	11	31.6	5.1
41448	829.5	46.1	11	11	75.4	4.2
60498	1746.3	75.1	11	11	158.8	6.8
84636	3446.6	136.0	11	11	313.3	12.4
114465	7000.1	259.4	12	12	583.3	21.6
150585	13058.8	431.0	12	12	1088.2	35.9
193596	24833.7	679.9	12	12	2069.5	56.7

a) The equation $\mathbf{B}\boldsymbol{\mu}_{\text{ind}} = \boldsymbol{\mathcal{F}}_{\text{mul}}$ was solved with a residual norm convergence threshold of 10^{-8} au. No parallelization was used.

b) Data reproduced from Ref. 140.

the largest system with almost 200k polarizable sites, FMM is more than 30 times faster than direct summation. FMM is already faster than the direct approach for the system with 4083 polarizable sites, and becomes more than 10 times faster for 41448 polarizable sites. The number of required iterations when using FMM or direct summation is identical. This indicates that FMM does not seem to affect the convergence behavior of the linear solvers, i.e., no numerical instabilities were observed. Lipparini shows a similar benchmark,[136] where the solution of the induced dipole moments for a system with 117k polarizable sites takes 22 seconds on 12 cores. A rough extrapolation to serial time would yield approximately 264 seconds, which is in the same range as the timings reported for the size-wise closest system with 114k polarizable sites. This comparison must be treated with caution, however, because the timings are recorded for different systems, different hardware, and different FMM implementations that do not share the same tunable parameters. The FMM code by Lipparini uses a different boxing scheme for near/far field interactions and is based on spherical multipoles. Nevertheless, the presented timings are somewhat similar to existing implementations, even though this is only a crude qualitative assessment.

6.2.2 Linear Scaling Test for SCF and Linear Response

Next, the linear scaling behavior in practical quantum chemical calculations is evaluated. The use of FMM is beneficial for systems where the iterative evaluation of induced fields becomes the rate-limiting step, i.e., the classical part of the calculation cost dominates the quantum part. To this end, I recorded timings for SCF and LR computations on the pNA systems with direct summation and with FMM ($p = 5, \theta = 0.5$). For these environment-dominated systems, it is expected that the wall times for SCF and LR procedures increase quadratically with the number of sites for direct summation, and FMM must show an asymptotic linear scaling. The recorded timings are shown in Figure 6.3. First, for SCF and LR with direct summation schemes, a quadratic increase in wall time is observed. Second, employing FMM mitigates the quadratic scaling and reduces the run time for larger systems significantly. A robust linear scaling is observed for environments with up to 200,000 polarizable sites. It is interesting to see the rather early onset of the quadratic function for SCF and LR jobs for systems with less than 100,000 atoms. This shows that the effect of FMM can be already beneficial for environments that are commonly used in PE computations. The cross-over point between the total run time of the direct and FMM implementations is found at 1440 sites. For the systems with more than 100,000 atoms,

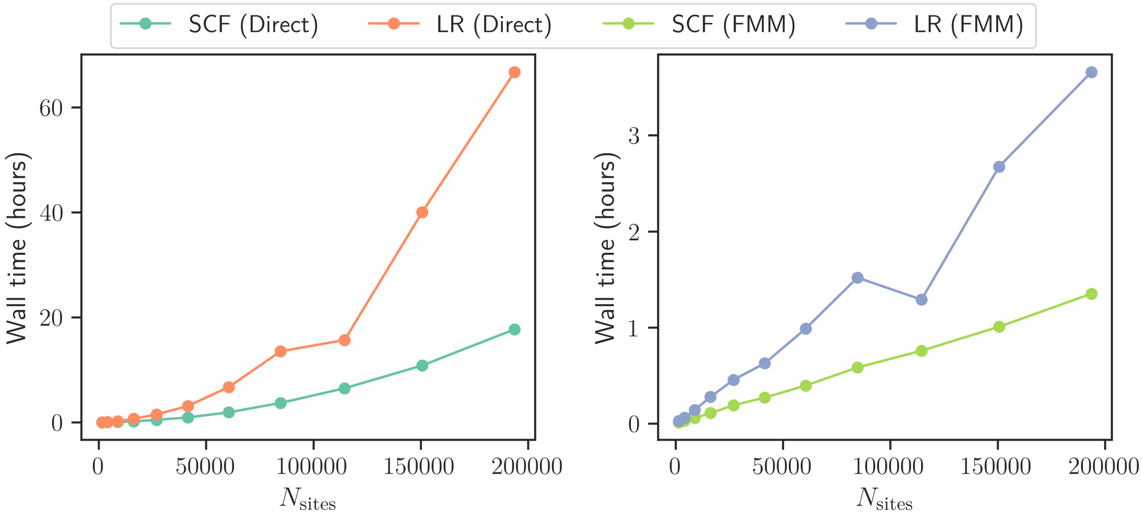


Figure 6.3: Scaling of direct (left) and FMM summation schemes (right). Wall times for CAM-B3LYP/6-31G(d,p) SCF and linear response computations are shown. The systems consist of box with a single pNA molecule surrounded by water molecules, consisting of N_{sites} atoms. Note that the vertical axes scales differ due to the large difference between direct and FMM wall times. Data reproduced from Ref. 140.

the SCF takes at least three times longer with direct summation than with FMM. It is thus advisable to use FMM for efficient PE computations already when moderately sized environments are used.

6.2.3 Accuracy of Electronic Excitation Spectra

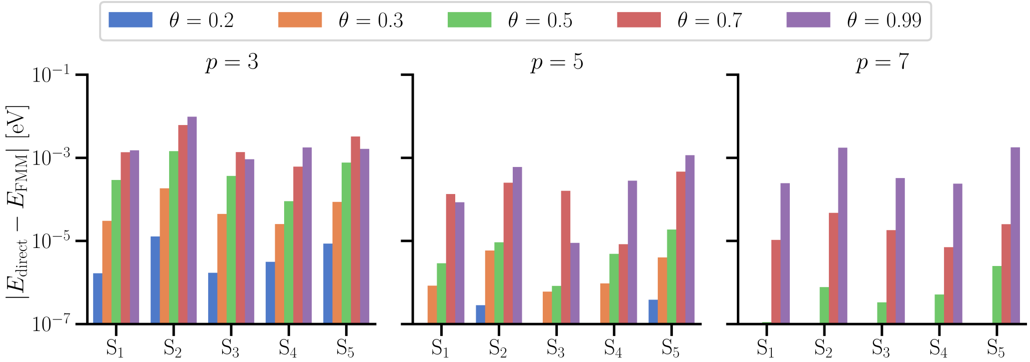


Figure 6.4: Errors in the excitation energies for pNA/38k H_2O computed with pt-PE-ADC(2). Note that with some parameter combinations, the FMM result is identical to the direct reference in all printed digits, in which case no bar is shown. Data reproduced from Ref. 140.

More important than the error of single field evaluations is the propagation of the FMM error to molecular properties, such as excitation energies and transition moments. Since one of the main focuses of this thesis is computational spectroscopy, I analyzed how the FMM error affects pt-PE-ADC(2) excitation energies. For this task, I used the pNA/38k H_2O test system, and the errors of the excitation energies of the five energetically lowest singlet states are shown in Figure 6.4. Since the energy convergence threshold in the eigensolver for these computations is on the order of 10^{-5} au, all errors below this threshold are numerically equal to zero in this error analysis. For $p = 3$, mean errors are on the order of 0.5 meV and the maximum error is 9.7 meV for pt-PE-ADC(2). A good trade-off in the accuracy is found for $p = 5$ and $\theta = 0.5$, where the excitation energy errors in this case are in the range between 10^{-5} and 10^{-3} eV. For practical calculations, i.e., analysis and plotting of excitation spectra, these errors are 'invisible'. The most important conclusion from this error analysis is that *all* errors are well below the intrinsic error of the employed ADC(2) method.[35,148,149] This is encouraging because it allows one to use PE in combination with FMM almost in a black-box manner for excited state calculations when reasonable defaults for the tree parameters are set (e.g., $p = 5$ and $\theta = 0.5$). FMM can be employed by users not familiar with the underlying principles without reducing the quality of the obtained results. For higher-order properties, however, one should again benchmark the FMM parameters, because larger deviations might be found if more and more dependent induced moment evaluations are coupled together.

6.2.4 Sizable Biomolecular Systems: Nile Red in BLG

To showcase the robustness of the FMM implementation, from a timing and memory perspective, I ran excited state computations with pt-PE-ADC(2) on the Nile Red molecule embedded in BLG and solvated by water. Due to the enormous amount of field integral evaluations, which were previously all stored in memory in PySCF, I had to refactor the interface code and implement a batching scheme for multipole potential and electric field integrals. The code automatically determines how much memory is available, estimates the memory use for PE-related integrals and then splits up the computation of integrals in an appropriate amount of individual batches of environment sites. An overview of the systems is presented in Figure 6.5. The system size was artificially increased for demonstration purposes by replicating the unit cell of the MD simulation box successively in all directions (Tab. 6.1). The timings for individual computational tasks and resulting spectra are presented in Figure 6.6.

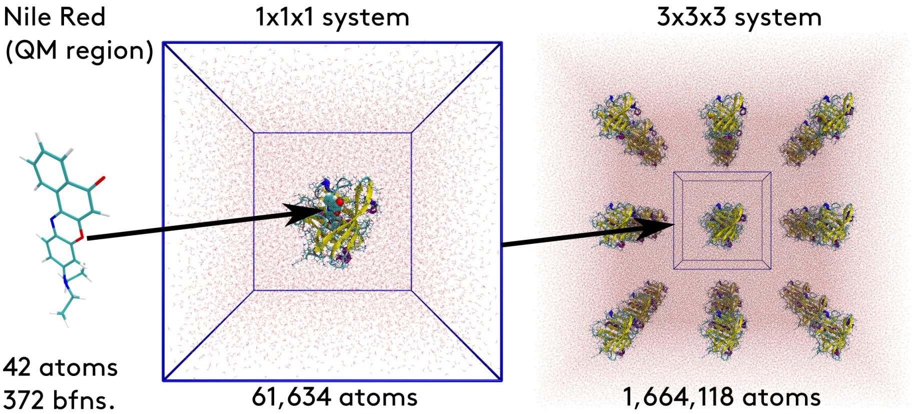


Figure 6.5: Illustration of Nile Red/BLG systems. Left: Nile Red molecule (quantum region). Center: single unit cell containing water, protein, and ions. Right: 27 unit cells (3x3x3) artificially replicated environment. The illustration was created using VMD[150] by P. Reinholdt.[1]

The largest system (3x3x3) contains more than 1.6 million polarizable sites in the environment. In the pt-PE-ADC(2) calculations, the number of polarizable sites only affects the run time of the SCF and the perturbative corrections (PTC), i.e., the ptSS and ptLR correction together. The ADC eigenvalue problem is solved without direct environment coupling, such that it always requires the same computational cost. Linearly ideal timing increases are 3-fold, 9-fold, and 27-fold for the 3x1x1, 3x3x1, and 3x3x3 system, respectively, when compared to the single unit cell 1x1x1 system. This trend is present for the SCF timings. Even though the observed relative timing factors are a bit larger than the theoretical estimates, the implementation is still capable of treating the largest systems in a linearly scaling manner. For PTC, the relative timing factors are below the theoretical estimate due to the fact that the number of iterations needed to solve for the induced moments varies a bit, especially when comparing to the smallest system. With this example, the ecological and sustainable aspect of using FMM for large polarizable environments becomes evident. Running these computations on the same hardware with direct summation would roughly take 100 days. Now, one could think about distributing the direct summation code over several nodes. To reach the speed of FMM with such a hypothetical code, it would require 100 nodes just to run the field evaluations, consuming 100-fold more power than the FMM code. Most importantly, the FMM implementation is robust with respect to system size, and no unexpected bottlenecks arise when treating such large systems. Timing analyses clearly show that the implementation is ready to handle polarizable environments of any practically relevant size without difficulties.

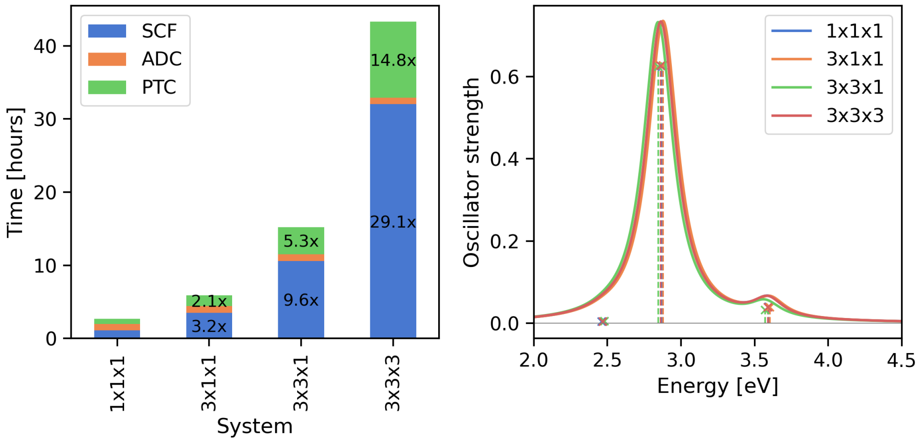


Figure 6.6: Timings and excitation spectra for Nile Red/BLG computed with pt-PE-ADC(2) employing FMM. Data reproduced from Ref. 140.

The observed eigenstates of the systems are expected to vary only slightly due to the long-range nature of the interactions with additional unit cells. This is indeed the case for all presented computations on Nile Red/BLG. For pt-PE-ADC(2), no direct coupling to the polarizable environment is taken into account in the ADC procedure, such that intensities are virtually identical in each system. The excitation energies are almost independent of the environment size. This is expected because the excitation energies should already be well converged for the smallest system, the larger ones are just artificially increased.

Consequently, the FMM implementation in CPPE can be employed for arbitrarily large polarizable environments, and yields efficient linear scaling and physically sound molecular properties.

6.3 Conclusions

In this chapter, I presented a linear-scaling formulation of the PE model, together with performance and accuracy analyses. Using FMM, the asymptotic scaling of the PE calculations with the number of sites in the environment is linear, and large speed-ups compared to direct summation techniques already for systems with a few thousand polarizable sites are observed. The required linear asymptotic scaling is proven with an extreme case study with more than one million polarizable environment sites. The implementation is openly available as part of the CPPE library,[79] and was published as a joint effort with an implementation in PElib.[140] The method allows for an accurate evaluation of the electric fields needed for the solution of the induced dipole polarization equations. FMM can be safely applied with almost no degradation to the accuracy of practical calculations of electronic excitation energies. A logical next step would be to improve the performance of PE calculations by reducing the computational effort of field integral evaluations for large systems by, e.g., efficient screening techniques. With the FMM implementation available in several of open-source quantum chemistry program packages via CPPE, it is now possible to model polarizable environments of virtually any size with a diverse set of density functional and wave function methods to target a multitude of molecular properties.

Notes

1. Reprinted with permission from: M. Scheurer, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted, “Efficient Open-Source Implementations of Linear-Scaling Polarizable Embedding: Use Octrees to Save the Trees”, *J. Chem. Theory Comput.* **2021**, *17 (6)*, 3445–3454. Copyright 2021 American Chemical Society.

Chapter 7

ADC/ISR Response Properties: Algorithms and Applications

In this chapter, I focus on describing algorithms to efficiently solve response equations in the ADC/ISR framework. The underlying theoretical framework was shown in Section 2.4, and the implementation of the modular response library `respondo` was explained in Section 4.3. After outlining the basic strategies to solve general response expressions with ADC/ISR, I show benchmark and example calculations of complex excited state polarizabilities (see Section 2.4.3 for the derivation), which is the first implementation of excited state response properties with ADC.[41] Finally, I propose an automated approach for convenient implementation of *any* response property based on SOS expressions.

7.1 Prerequisites

Recall the general, complex response equation (2.99). The following discussion focuses on this more general case with complex frequency arguments, and the equations for real equations can be easily obtained by setting $\gamma = 0$, i.e., discarding everything but real blocks of the matrices. As an initial step, the response vector consists of a real and an imaginary part

$$\mathbf{x} = \mathbf{x}^R + i\mathbf{x}^I. \tag{7.1}$$

Parts of this chapter have already been published in:

- M. Scheurer, T. Fransson, P. Norman, A. Dreuw, and D. R. Rehn, "Complex Excited State Polarizabilities in the ADC/ISR Framework", *J. Chem. Phys.* **2020**, *153*, 074112. (Reference 41)

The right-hand side of the linear equation is partitioned accordingly. To avoid complex algebra, the general response equation is recast to a form of double dimension, i.e., the real and imaginary part of the vector are vertically stacked.[41,54,56,151] Carrying out the matrix multiplication in (2.99), we have

$$\begin{aligned}(\mathbf{M} - \omega - i\gamma)\mathbf{x} &= \mathbf{M}\mathbf{x} - \omega\mathbf{x} - i\gamma\mathbf{x} \qquad (7.2)\\ &= \mathbf{M}\mathbf{x}^R - \omega\mathbf{x}^R - i\gamma\mathbf{x}^R + i\mathbf{M}\mathbf{x}^I - i\omega\mathbf{x}^I + \gamma\mathbf{x}^I. \qquad (7.3)\end{aligned}$$

Sorting real and imaginary parts, one arrives at the following block matrix form

$$\begin{pmatrix}\mathbf{M} - \omega & \gamma \\ -\gamma & \mathbf{M} - \omega\end{pmatrix}\begin{pmatrix}\mathbf{x}^R \\ \mathbf{x}^I\end{pmatrix} = \begin{pmatrix}\boldsymbol{\mathcal{R}}^R \\ \boldsymbol{\mathcal{R}}^I\end{pmatrix}, \qquad (7.4)$$

which is non-Hermitian. By multiplication with $\begin{pmatrix}1 & 0 \\ 0 & -1\end{pmatrix}$ from the left, the Hermitian form is obtained

$$\begin{pmatrix}\mathbf{M} - \omega & \gamma \\ \gamma & -(\mathbf{M} - \omega)\end{pmatrix}\begin{pmatrix}\mathbf{x}^R \\ \mathbf{x}^I\end{pmatrix} = \begin{pmatrix}\boldsymbol{\mathcal{R}}^R \\ -\boldsymbol{\mathcal{R}}^I\end{pmatrix}, \qquad (7.5)$$

which is more suitable for common iterative solvers. Note that one has to take care of the negative sign of the imaginary part of the right-hand side. This block is, however, zero for real-valued operators, e.g., in case the right-hand side is equal to a modified transition moment obtained from the electric dipole operator. To solve eq (7.5) numerically, almost any linear solver can be employed. Since the system of equations is linear and symmetric, a conjugate gradient (CG) solver[152] is employed by default in `respondo`, for example. Convergence is improved by preconditioning the residual vectors, leading to a preconditioned CG (PCG) algorithm. The simplest preconditioner **P** in this case is the Jacobi preconditioner using the diagonal of the problem matrix. The preconditioner matrix **P** can be formed by inverting the block matrix

$$\begin{aligned}\mathbf{P} &= \begin{pmatrix}\mathbf{D} - \omega & \gamma \\ \gamma & -(\mathbf{D} - \omega)\end{pmatrix}^{-1} \qquad (7.6)\\ &= \frac{1}{-(\mathbf{D} - \omega)^2 - \gamma^2}\begin{pmatrix}-(\mathbf{D} - \omega) & -\gamma \\ -\gamma & \mathbf{D} - \omega\end{pmatrix} \qquad (7.7)\end{aligned}$$

with the diagonal **D** of the ADC matrix **M**. These ingredients suffice to iteratively solve eq (7.5) with standard numerical techniques. There are, however, other strategies to be applied to solve the CPP linear equations, which have proven successful for other methods than ADC as well.[56,151] One option is to explicitly solve the CPP response equations in a reduced space,[151] the corresponding algorithmic details for ADC are summarized in Section 7.2.1. This solver algorithm is applicable to all ADC methods. Another option for the second-order ADC(2) method exploits that the `pphh-pphh` block of the ADC(2) is a diagonal matrix, i.e., it is easily invertible. With the Schur complement of the ADC matrix, the `pphh` part of the solution vector can be folded into the `ph` block, reducing the computational cost and improving convergence. The corresponding algorithm is described in Section 7.2.2. In the following equations, the matrix blocks are denoted by subscripts **1** and **2** for `ph` and `pphh` parts, respectively, for clarity. Inserting all blocks of the ADC(2) secular matrix into eq (7.5) yields

$$\begin{pmatrix} \mathbf{M}_{\mathbf{11}} - \omega & \mathbf{M}_{\mathbf{12}} & \gamma & 0 \\ \mathbf{M}_{\mathbf{21}} & \mathbf{D}_{\mathbf{22}} - \omega & 0 & \gamma \\ \gamma & 0 & -\left(\mathbf{M}_{\mathbf{11}} - \omega\right) & -\mathbf{M}_{\mathbf{12}} \\ 0 & \gamma & -\mathbf{M}_{\mathbf{21}} & -\left(\mathbf{D}_{\mathbf{22}} - \omega\right) \end{pmatrix} \begin{pmatrix} \mathbf{x}_{\mathbf{1}}^{R} \\ \mathbf{x}_{\mathbf{1}}^{I} \\ \mathbf{x}_{\mathbf{2}}^{R} \\ \mathbf{x}_{\mathbf{2}}^{I} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{R}}_{\mathbf{1}}^{R} \\ \boldsymbol{\mathcal{R}}_{\mathbf{2}}^{R} \\ -\boldsymbol{\mathcal{R}}_{\mathbf{1}}^{I} \\ -\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{I} \end{pmatrix} \tag{7.8}$$

From eq (7.8) one extracts expressions for the `pphh` parts of the response vector

$$\mathbf{x}_{\mathbf{2}}^{R} = \frac{1}{\left(\mathbf{D}_{\mathbf{22}} - \omega\right)^{2} + \gamma^{2}} \left[\left(\mathbf{D}_{\mathbf{22}} - \omega\right)\left(\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{R} - \mathbf{M}_{\mathbf{21}}\mathbf{x}_{\mathbf{1}}^{R}\right) - \gamma\left(\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{I} - \mathbf{M}_{\mathbf{21}}\mathbf{x}_{\mathbf{1}}^{I}\right)\right] \tag{7.9}$$

$$\mathbf{x}_{\mathbf{2}}^{I} = \frac{1}{\left(\mathbf{D}_{\mathbf{22}} - \omega\right)^{2} + \gamma^{2}} \left[\left(\mathbf{D}_{\mathbf{22}} - \omega\right)\left(\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{I} - \mathbf{M}_{\mathbf{21}}\mathbf{x}_{\mathbf{1}}^{I}\right) + \gamma\left(\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{R} - \mathbf{M}_{\mathbf{21}}\mathbf{x}_{\mathbf{1}}^{R}\right)\right], \tag{7.10}$$

that is, the `ph` and `pphh` parts of the response vector are decoupled. This can be used to rewrite the system of linear equations in the `ph` space

$$\begin{pmatrix} \mathbf{M}'_{\mathbf{11}} - \mathbf{M}_{\mathbf{12}}\frac{\mathbf{D}'_{\mathbf{22}}}{\mathbf{D}'_{\mathbf{22}}{}^{2}+\gamma^{2}}\mathbf{M}_{\mathbf{21}} & \mathbf{M}_{\mathbf{12}}\frac{\gamma}{\mathbf{D}'_{\mathbf{22}}{}^{2}+\gamma^{2}}\mathbf{M}_{\mathbf{21}} + \gamma \\ \mathbf{M}_{\mathbf{12}}\frac{\gamma}{\mathbf{D}'_{\mathbf{22}}{}^{2}+\gamma^{2}}\mathbf{M}_{\mathbf{21}} + \gamma & -\left(\mathbf{M}'_{\mathbf{11}} - \mathbf{M}_{\mathbf{12}}\frac{\mathbf{D}'_{\mathbf{22}}}{\mathbf{D}'_{\mathbf{22}}{}^{2}+\gamma^{2}}\mathbf{M}_{\mathbf{21}}\right) \end{pmatrix} \begin{pmatrix} \mathbf{x}_{\mathbf{1}}^{R} \\ \mathbf{x}_{\mathbf{1}}^{I} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{R}}_{\mathbf{1}}^{\prime R} \\ \boldsymbol{\mathcal{R}}_{\mathbf{1}}^{\prime I} \end{pmatrix}, \tag{7.11}$$

with $\mathbf{M}'_{\mathbf{11}} = \mathbf{M}_{\mathbf{11}} - \omega$ and $\mathbf{D}'_{\mathbf{22}} = \mathbf{D}_{\mathbf{22}} - \omega$. The `ph`-space right-hand-side is found as

$$\boldsymbol{\mathcal{R}}_{\mathbf{1}}^{\prime R} = \boldsymbol{\mathcal{R}}_{\mathbf{1}}^{R} - \mathbf{M}_{\mathbf{12}}\frac{\mathbf{D}'_{\mathbf{22}}}{\mathbf{D}'_{\mathbf{22}}{}^{2} + \gamma^{2}}\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{R} + \mathbf{M}_{\mathbf{12}}\frac{\gamma}{\mathbf{D}'_{\mathbf{22}}{}^{2} + \gamma^{2}}\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{I} \tag{7.12}$$

$$\boldsymbol{\mathcal{R}}_{\mathbf{1}}^{\prime I} = -\left(\boldsymbol{\mathcal{R}}_{\mathbf{1}}^{I} - \mathbf{M}_{\mathbf{12}}\frac{\mathbf{D}'_{\mathbf{22}}}{\mathbf{D}'_{\mathbf{22}}{}^{2} + \gamma^{2}}\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{I} + \mathbf{M}_{\mathbf{12}}\frac{\gamma}{\mathbf{D}'_{\mathbf{22}}{}^{2} + \gamma^{2}}\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{R}\right). \tag{7.13}$$

The reduced matrix and right-hand side equations can be directly used in any iterative solver procedure, decreasing the size of the problem matrix dramatically. As a result, less memory is required and the equations converge much faster by only computing the residual of the `pphh` block indirectly. Once the `ph` part of the vectors is converged, the exact `pphh` solutions are computed using eqs (7.9) and (7.10).

7.2 Solver Algorithms

This section describes algorithms to solve complex response equations within the ADC/ISR approach relying on a subspace projection technique. The subspace algorithm has been used primarily for HF and DFT response properties in the past.[151] Here, I show a comprehensive description of this CPP solver algorithm for ADC. Note that for SCF response properties, the vectors do not have a `pphh` block, such that the memory bottleneck is much lower than for ADC. This makes simultaneous solution for multiple frequencies and/or right-hand sides difficult. The response equations are then solved for each frequency and right-hand side individually to avoid expensive memory accumulation.

7.2.1 CPP Solver Algorithm

Guess Vectors Initial guesses for the linear solver procedure are formed by applying the (Jacobi) preconditioner matrix $\mathbf{P}$ to the right-hand side vector $\boldsymbol{\mathcal{R}}$, i.e.,

$$\mathbf{g} = \mathbf{P}\boldsymbol{\mathcal{R}}. \tag{7.14}$$

Next, the real and imaginary parts of $\mathbf{g}$ are split up and taken as the initial subspace vectors $\{\mathbf{x}_n\}$. As such, more flexibility due to mixing real and imaginary parts is possible.

Iterations The iterative solver proceeds as follows:

1. Compute matrix applies for new subspace vectors:

$$\{\mathbf{s}_n\} = \{\mathbf{M}\mathbf{x}_1, \ldots, \mathbf{M}\mathbf{x}_n\} \tag{7.15}$$

2. Compute the subspace projection of $\mathbf{M}$

$$M_{ij}^{\text{sub}} = \mathbf{x}_i^T \mathbf{M}\mathbf{x}_j = \mathbf{x}_i^T \mathbf{s}_j \ , \tag{7.16}$$

and subtract the frequency ω

$$\mathbf{M}^{\omega,\text{sub}} = \mathbf{M}^{\text{sub}} - \omega \ . \tag{7.17}$$

Compute subspace projections of the right-hand side

$$\mathcal{R}_i^{\text{sub},R} = \mathbf{x}_i^T \boldsymbol{\mathcal{R}}^R \tag{7.18}$$

$$\mathcal{R}_i^{\text{sub},I} = \mathbf{x}_i^T \boldsymbol{\mathcal{R}}^I \tag{7.19}$$

3. Construct the linear problem in the reduced space

$$\begin{pmatrix} \mathbf{M}^{\omega,\text{sub}} & \gamma \\ \gamma & -\mathbf{M}^{\omega,\text{sub}} \end{pmatrix} \begin{pmatrix} \mathbf{x}^{\text{sub},R} \\ \mathbf{x}^{\text{sub},I} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{R}}^{\text{sub},R} \\ \boldsymbol{\mathcal{R}}^{\text{sub},I} \end{pmatrix} \tag{7.20}$$

and solve for $\mathbf{x}^{\text{sub}}$. The current optimal solution in the full space is given by

$$\mathbf{x}^{\text{opt},R} = \sum_{i=1}^{n} x_i^{\text{sub},R} \mathbf{x}_i \tag{7.21}$$

$$\mathbf{x}^{\text{opt},I} = \sum_{i=1}^{n} x_i^{\text{sub},I} \mathbf{x}_i \tag{7.22}$$

4. Compute the residual vector

$$\mathbf{r} = \begin{pmatrix} \mathbf{M} - \omega & \gamma \\ \gamma & -(\mathbf{M} - \omega) \end{pmatrix} \begin{pmatrix} \mathbf{x}^{\text{opt},R} \\ \mathbf{x}^{\text{opt},I} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\mathcal{R}}^R \\ \boldsymbol{\mathcal{R}}^I \end{pmatrix} , \tag{7.23}$$

reusing the cached vectors as

$$\mathbf{r} = \begin{pmatrix} \sum_{i=1}^{n} x_i^{\text{sub},R} \mathbf{s}_i - \omega x_i^{\text{sub},R} + \gamma x_i^{\text{sub},I} \\ \sum_{i=1}^{n} -x_i^{\text{sub},I} \mathbf{s}_i + \omega x_i^{\text{sub},I} + \gamma x_i^{\text{sub},R} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\mathcal{R}}^R \\ \boldsymbol{\mathcal{R}}^I \end{pmatrix} . \tag{7.24}$$

5. Convergence check, preconditioning, subspace collapse:
   - If the residual norm $|\mathbf{r}|$ smaller than convergence threshold, form solution, exit.
   - Else precondition the residual vector $\mathbf{p} = \mathbf{P}\mathbf{r}$ and add its real and imaginary part to $\{\mathbf{x}_n\}$ while orthogonalizing against the existing vectors. If the number of subspace vectors is greater than the maximum number, collapse the subspace

by computing the current optimal solution $\mathbf{x}^{\mathrm{opt}}$, adding its real and imaginary part to the subspace and discarding all other previous subspace vectors. Start from step 1.

7.2.2 Single-Space CPP Solver Algorithm

Guess Vectors For the single-excitation (ph) space solver, the following preconditioner matrix is used

$$\mathbf{P}' = \frac{1}{-(\mathbf{D_{11}} - \omega)^2 - \gamma^2} \begin{pmatrix} -(\mathbf{D_{11}} - \omega) & -\gamma \\ -\gamma & \mathbf{D_{11}} - \omega \end{pmatrix} \tag{7.25}$$

with the diagonal $\mathbf{D_{11}}$ of the ph-ph block of $\mathbf{M}$.

Iterations As for the previous algorithm, real and imaginary parts share a single subspace $\{\mathbf{x}'_n\}$. A little more bookkeeping is required for the singles-space algorithm though. For convenience, one can collect and rename some matrix blocks, i.e.,

$$\begin{pmatrix} \mathbf{M_{11}} - \omega - \mathbf{M_{12}} \frac{\mathbf{D}'_{\mathbf{22}}}{{\mathbf{D}'_{\mathbf{22}}}^2 + \gamma^2} & \mathbf{M_{12}} \frac{\gamma}{{\mathbf{D}'_{\mathbf{22}}}^2 + \gamma^2} + \gamma \\ \mathbf{M_{12}} \frac{\gamma}{{\mathbf{D}'_{\mathbf{22}}}^2 + \gamma^2} + \gamma & -\left(\mathbf{M_{11}} - \omega - \mathbf{M_{12}} \frac{\mathbf{D}'_{\mathbf{22}}}{{\mathbf{D}'_{\mathbf{22}}}^2 + \gamma^2}\right) \end{pmatrix} = \begin{pmatrix} \mathbf{M_{11}} - \mathbf{S} & \mathbf{G} \\ \mathbf{G} & -(\mathbf{M_{11}} - \mathbf{S}) \end{pmatrix} \tag{7.26}$$

1. Compute matrix applies and cache results
   - Apply $\mathbf{M_{11}} - \mathbf{S}$:

     $$\{\mathbf{m}'_n\} = \{(\mathbf{M_{11}} - \mathbf{S})\mathbf{x}'_1, \ldots, (\mathbf{M_{11}} - \mathbf{S})\mathbf{x}'_n\} \tag{7.27}$$

   - Apply $\mathbf{G}$

     $$\{\mathbf{g}_n\} = \{\mathbf{G}\mathbf{x}'_1, \ldots, \mathbf{G}\mathbf{x}'_n\} \tag{7.28}$$

2. Compute subspace projections of $\mathbf{M_{11}} - \mathbf{S}$

   $$M^{\mathrm{sub}}_{ij} = \mathbf{x}'^T_i \mathbf{m}'_j \tag{7.29}$$

   and $\mathbf{G}$

   $$G^{\mathrm{sub}}_{ij} = \mathbf{x}'^T_i \mathbf{g}_j \tag{7.30}$$

The projected right-hand side $\boldsymbol{\mathcal{R}}'^{\mathrm{sub}}$ can be computed analogously to the conventional solver.

3. Construct the linear problem in the reduced space

$$\begin{pmatrix} \mathbf{M}^{\mathrm{sub}} & \mathbf{G}^{\mathrm{sub}} \\ \mathbf{G}^{\mathrm{sub}} & -\mathbf{M}^{\mathrm{sub}} \end{pmatrix} \begin{pmatrix} \mathbf{x}'^{\mathrm{sub},R} \\ \mathbf{x}'^{\mathrm{sub},I} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{R}}'^{\mathrm{sub},R} \\ \boldsymbol{\mathcal{R}}'^{\mathrm{sub},I} \end{pmatrix} \tag{7.31}$$

and solve for $\mathbf{x}'^{\mathrm{sub}}$. The current optimal solution (in the singles space only) is given by

$$\mathbf{x}'^{\mathrm{opt},R} = \sum_{i=1}^{n} x_i'^{\mathrm{sub},R} \mathbf{x}_i' \tag{7.32}$$

$$\mathbf{x}'^{\mathrm{opt},I} = \sum_{i=1}^{n} x_i'^{\mathrm{sub},I} \mathbf{x}_i' \tag{7.33}$$

4. Compute the residual vector

$$\mathbf{r} = \begin{pmatrix} \mathbf{M_{11}} - \mathbf{S} & \mathbf{G} \\ \mathbf{G} & -(\mathbf{M_{11}} - \mathbf{S}) \end{pmatrix} \begin{pmatrix} \mathbf{x}'^{\mathrm{opt},R} \\ \mathbf{x}'^{\mathrm{opt},I} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\mathcal{R}}'^{R} \\ \boldsymbol{\mathcal{R}}'^{I} \end{pmatrix} \tag{7.34}$$

using the intermediates from above

$$\mathbf{r} = \begin{pmatrix} \sum_{i=1}^{n} x_i'^{\mathrm{sub},R} \mathbf{m}_i' + x_i'^{\mathrm{sub},I} \mathbf{g}_i \\ \sum_{i=1}^{n} -x_i'^{\mathrm{sub},I} \mathbf{m}_i' + x_i'^{\mathrm{sub},R} \mathbf{g}_i \end{pmatrix} - \begin{pmatrix} \boldsymbol{\mathcal{R}}'^{R} \\ \boldsymbol{\mathcal{R}}'^{I} \end{pmatrix}. \tag{7.35}$$

5. Convergence check, preconditioning, subspace collapse:

   - If the residual norm $|\mathbf{r}|$ smaller than convergence threshold, compute the solutions in the singles space, then compute the doubles space solutions according to eqs 7.9 and 7.10.
   - Else precondition the residual vector $\mathbf{p} = \mathbf{P}'\mathbf{r}$ and add its real and imaginary part to $\{\mathbf{x}_n'\}$ while orthogonalizing against the existing vectors. If the number of subspace vectors is greater than the maximum number, collapse the subspace by computing the current optimal solution $\mathbf{x}'^{\mathrm{opt}}$, adding its real and imaginary part to the subspace and discarding all other previous subspace vectors. Start from step 1.

7.2.3 Convergence Analysis of PCG and CPP Solvers

To illustrate how different solvers and the folding of the `pphh` part of the matrix affect convergence of the linear equations in practice, I carried out minimal test calculations on the pyridine molecule using ADC(2)/aug-cc-pVDZ[128] with an optimized geometry obtained with MP2/cc-pVTZ as implemented in Q-Chem.[93,94] The pyridine molecule was placed in the xz plane. For brevity, only the y-component of the modified transition moment with the electric dipole operator $\mathbf{F}(\hat{\mu}_y)$ was used as the right-hand side to solve a) the equation for the static polarizability ($\mathbf{Mx} = \mathbf{F}(\hat{\mu}_y)$) and b) for the complex polarizability evaluated at the frequency of the energetically lowest singlet excited state ($(\mathbf{M} - \omega_1 - i\gamma)\mathbf{x} = \mathbf{F}(\hat{\mu}_y)$). In case b), a damping parameter $\gamma = 10^{-4}$ au was used. Both equations were then solved using either a PCG algorithm or the CPP solver, once in the full space and once in the single-excitation space only (folded `pphh` part), as implemented in `adcc` and `respondo` with PySCF as host program for the SCF.[39] The convergence analysis, i.e., residual norm in each iteration of each solver/folding combination is shown in Figure 7.1. Clearly, the simple matrix inversion does not show any convergence issues, as expected. The folded versions require four iterations less to achieve convergence than the full matrix counterpart. From a performance point of view, it is thus advisable to use the PCG solver directly, because only a single matrix apply needs to be computed per iteration. On the contrary, solving the response equation at the electronic resonance in

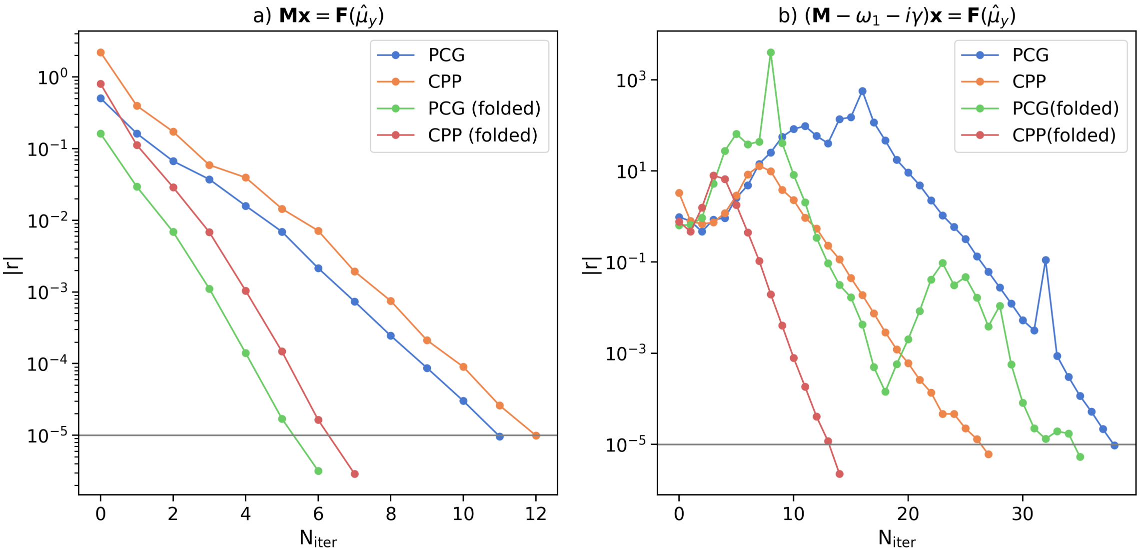


Figure 7.1: Convergence analysis for a) $\mathbf{Mx} = \mathbf{F}(\hat{\mu}_y)$ and b) $(\mathbf{M} - \omega_1 - i\gamma)\mathbf{x} = \mathbf{F}(\hat{\mu}_y)$ for the pyridine molecule at the ADC(2)/aug-cc-pVDZ level of theory using folded and unfolded matrices with the PCG and CPP solvers.

case b) proves to be difficult for all solver and matrix versions. Especially the PCG solver needs roughly 40 iterations to achieve convergence, even with the folded ADC(2) matrix, 35 iterations are needed. The CPP solver is in this case more stable, requiring less than 30 iterations to converge the response vector. In case the CPP solver is used in combination with the folded ADC(2) matrix, convergence is reached in just 14 iterations, without any large jumps in the residual norm as seen for the other combinations. By mixing real and imaginary parts of the residual vectors in the subspace, a rather stable progression to convergence is observed. Furthermore, by reduction of the problem size to the `ph` space only, large erroneous contributions to the `pphh` part of the vector are not a problem anymore. This problem becomes even more pronounced when solving response equations in the X-ray regime of the excitation spectrum, and can often only be remedied by further approximations.[153] The brief analysis above shows how useful the newly implemented solvers in `respondo` can be in practical simulations when a robust numerical procedure to solve response equations is required. For this reason, the solvers are of course publicly available in the `respondo` library.

7.3 Complex Excited State Polarizabilities

As explained previously, I evaluate the dipole polarizability of an excited state directly through its linear response function. An alternative route would be to compute the double residue of the ground state cubic response function.[154–156] With the linear response function of the excited state in combination with damped response theory[2] it is also possible to obtain to one-photon absorption cross-sections as well as C_6 dispersion coefficients for excited states. To this end, I implemented complex excited state polarizabilities in the `respondo` library using the `adcc` API for all ADC-related working equations. As for all other response properties in `respondo`, complex excited state polarizabilities are available for all ADC methods accessible via ADC, that is, also ADC(3).

Other quantum chemical methods have been previously employed to compute frequency-dependent excited state polarizabilities, e.g., Hartree-Fock,[154,157–159] coupled cluster,[155,156,160–163] multi-reference approaches,[158,159,161,164] and DFT methods.[163,165–167] Calculations of excited state C_6 dispersion coefficients are more scarce[166] and this property is also difficult to determine experimentally. The common strategy for ADC response properties is to exploit the ISR to resolve SOS expressions such that they become computationally accessible. In addition to this expectation-value-based strategy, it is possible to define a response property in terms of derivatives of the energy.[69] A comparative analysis of these approaches was recently conducted for ADC methods.[69] For excited state polarizabilities, both approaches have been reported for equation-of-motion coupled cluster with singles and doubles (EOM-CCSD).[156,168–170] The ISR-based ansatz described here is comparable to the expectation-value coupled cluster approach to molecular properties, and both methods will be analyzed and compared to experimental data where available. Next, the implementation and computational methodology is presented, followed by the results of test calculations on s-tetrazine, pyrimidine, formaldehyde, naphthalene, uracil, and p-nitroaniline for static excited state polarizabilities, and computations of excited state C_6 dispersion coefficients for pyridine, pyrazine, and s-tetrazine.

7.3.1 Implementation

Only the matrix-vector products of the non-shifted ADC/ISR matrices with a trial vector $\mathbf{v}$ are available in `adcc` and the modified matrices $\mathbf{M}_f$ and $\mathbf{B}_f$ are implemented by projecting out all components along the eigenvector $\mathbf{x}_f$ after the matrix multiplication, i.e.,

$$\mathbf{M}_f\mathbf{v} = \mathbf{M}\mathbf{v} - \omega_f\mathbf{v} - \mathbf{x}_f\frac{\mathbf{x}_f^\dagger\mathbf{M}\mathbf{v}}{\mathbf{x}_f^\dagger\mathbf{x}_f}. \tag{7.36}$$

The projection scheme for the $\mathbf{B}_f$ matrix is analogous. As explained above, a CG algorithm with Jacobi preconditioning was used to obtain ground and excited state polarizabilities. The excited state polarizabilities were implemented in an early development version of `respondo` using the `adcc` toolkit for all ADC-based routines.[80] Since only the full matrix representation for $\mathbf{B}$ is given in Ref. 40, I derived the necessary matrix-vector product, and the programmable expressions are shown in the Appendix. At the time of the first implementation, the working equations were still written in closed-source C++ code. They can, however, now be easily implemented with Python-side `adcc` functionality. To test the implementation, eq (2.93) was evaluated for small systems (H_2O/6-31G, LiH/STO-3G), where a full ADC(2) matrix diagonalization computationally achievable. The results from evaluating the SOS expression were compared to the result from the linear solvers, and were found to agree up to numerical accuracy (data not shown).

7.3.2 Computational Details

Geometries for s-tetrazine, pyrimidine, uracil and p-nitroaniline (pNA) were obtained from Ref. 156. For s-tetrazine, static polarizabilities of the ground state and the $1^1\mathrm{B}_{1u}$ excited state were computed using the Sadlej-pVTZ basis set[171] and the geometry of the corresponding electronic state, as described in Ref. 156. Results were obtained using ADC(2), ADC(2)-x, and ADC(3), as now implemented in `adcc`. SCF results were obtained using PySCF.[39,95] In all ADC calculations, the second-order ISR was employed.[40,52] In combination with third-order ADC, this results in the ADC(3/2) approximation. For consistency, the calculations using the EOM-CCSD derivative and expectation-value approaches were repeated from Ref. 156, employing the Q-Chem 5.2 program package.[93,94] The same methods were employed to compute static polarizabilities of the pyrimidine ground state and the $1^1\mathrm{B}_2$ excited state. Formaldehyde and naphthalene were optimized at the MP2/cc-pVTZ level of theory using Q-Chem 5.2,[93,94] where the former was placed in the xz-plane and the latter in the xy-plane. For formaldehyde, the polarizabilities of the ground state and the $1^1\mathrm{B}_1$ excited state were computed, whereas the ground state and $1^1\mathrm{B}_{3\mathrm{u}}$ state polarizabilities were obtained for naphthalene, using all three ADC methods and CCSD with the aug-cc-pVDZ basis set. For uracil and pNA, the aug-cc-pVDZ basis set was employed to compute the polarizabilities of the ground state and $1^1\mathrm{A}''$ and $2^1\mathrm{A}'$, as well as $2^1\mathrm{A}_1$ excited states, respectively. For the computation of C_6 dispersion coefficients, pyridine, and pyrazine geometries were optimized at the MP2/cc-pVTZ level of theory using Q-Chem 5.2,[93,94] whereas the previously mentioned ground state geometry was taken for s-tetrazine. Ground and excited state C_6 dispersion coefficients for these three molecules were computed according to the procedure outlined in Ref. 52. For pyridine, the $1^1\mathrm{B}_2$ excited state was considered, and $1^1\mathrm{B}_{3\mathrm{u}}$ for pyrazine and s-tetrazine. Results were analyzed using `cclib`[172], `pandas`[112,113], and plotted using `matplotlib`[109] and `seaborn`.[173] The computed data, which are published in Ref. 41, are openly available on GitHub (`https://github.com/maxscheurer/adc_excipol_data`) and have been deposited on Zenodo with the DOI `10.5281/zenodo.3770508`.

7.3.3 Numerical Case Studies

Excited state polarizabilities and C_6 coefficients were computed for s-tetrazine, pyrimidine, uracil, and pNA. The static polarizabilities of these molecules were previously investigated using EOM-CCSD.[156] The ADC results are herein compared to this study, analyzing the ADC/ISR results at different levels of perturbation theory. In addition, static polarizabilities of formaldehyde and naphthalene are compared to experimental data and among the employed methodologies. Anticipated trends for excited state polarizabilities have been thoroughly discussed in Ref. 156: In brief, states with a large exciton size tend to have larger polarizabilities than the electronic ground state (e.g., Rydberg states), whereas the opposite should be the case for excited states with a large permanent dipole moment, e.g., CT states. From the SOS expression of the polarizability (eq (2.93)), it also becomes clear that low-lying dipole-allowed excited states should possess larger polarizabilities than the electronic ground state due to coupling to the ground state and to higher-lying excited states.[156] To discuss the different computational methods, absolute differences of isotropic polarizabilities are used, $\alpha_{\text{iso}} = \frac{1}{3}(\alpha_{xx} + \alpha_{yy} + \alpha_{zz})$. The largest differences can be expected in comparison with derivative-based CCSD (CCSD Der.), since this approach follows a different strategy than expectation-value-based methods. Thus, the deviation defined as $\delta_{\text{Der.}} = |\alpha_{\text{iso}}(\text{CCSD Der.}) - \alpha_{\text{iso}}(\text{expectation-value method})|$ is employed to discuss the results. Note that CCSD Der. results are not considered superior, i.e., they are not the theoretical best estimate for excited state polarizabilities in this section.

s-Tetrazine and Pyrimidine

Table 7.1: Static polarizabilities of the *s*-tetrazine ground and excited $1^1\text{B}_{1\text{u}}$ states.[a), b)]

State	ground state		$1^1\text{B}_{1\text{u}}$		
	$(\alpha_{xx},\, \alpha_{yy},\, \alpha_{zz})$	α_{iso}	$(\alpha_{xx},\, \alpha_{yy},\, \alpha_{zz})$	α_{iso}	E_{exc}
ADC(2)	(66.08, 61.26, 33.50)	53.61	(39.06, 78.55, 15.71)	44.44	2.20
ADC(2)-x	(69.38, 64.96, 35.72)	56.69	(50.64, 80.08, 21.60)	50.77	1.31
ADC(3/2)	(64.10, 59.57, 33.41)	52.36	(57.28, 74.94, 25.90)	52.70	2.18
CCSD Der.	(60.73, 56.02, 32.73)	49.83	(66.03, 80.09, 31.93)	59.35	2.39
CCSD E.V.	(62.82, 58.01, 33.48)	51.44	(49.28, 71.75, 23.05)	48.03	2.39

a) Data reproduced from Ref. 41.
b) Polarizability components (α_{AA}, α_{iso}) in au, excitation energies (E_{exc}) in eV.

Table 7.1 shows the Cartesian components of the static polarizabilities for the ground and $1^1\text{B}_{1\text{u}}$ states of *s*-tetrazine, as obtained using ADC and CCSD methods.[156] As previously stated, most CCSD results discussed herein have already been reported,[156] and have only been amended by CCSD expectation-value (E.V.) results for completeness.

For the ground state static polarizabilities, all methods yield results of comparable magnitude, with differences $\delta_{\text{Der.}}$ of 3.78, 6.86, 2.53, and 1.61 au for ADC(2), ADC(2)-x, ADC(3/2), and CCSD E.V., respectively. For excited state polarizabilities the differences are larger, with a decreasing discrepancy trend for the ADC hierarchy of 14.91 au for ADC(2), 8.58 au for ADC(2)-x, and 6.64 au for ADC(3/2). By comparison, the deviation for CCSD E.V. is 11.32 au. ADC(3/2) is thus in close agreement with the derivative-based EOM-CCSD result. An experimental result for the anisotropy of the polarizability ($\Delta\alpha = \frac{1}{2}(\alpha_{yy} + \alpha_{xx}) - \alpha_{zz}$) is reported as 5.4 and 45.2 au for the ground state and the lowest singlet state, respectively.[174] All computational methods overshoot the anisotropy for the electronic ground state (30.2, 31.5, 28.4, 25.6, and 26.9 au for ADC(2), ADC(2)-x, ADC(3/2), CCSD Der., and CCSD E.V. respectively). However, the anisotropy of the lowest singlet state static polarizability match the experimental result reasonably well (43.1, 43.8, 40.2, 41.1, and 37.5 au for ADC(2), ADC(2)-x, ADC(3/2), CCSD Der., and CCSD E.V. respectively).

Table 7.2: Static polarizabilities of the pyrimidine ground state and $1^1\mathrm{B}_2$.[a), b)]

State	ground state		$1^1\mathrm{B}_2$		
	(α_{xx}, α_{yy}, α_{zz})	α_{iso}	(α_{xx}, α_{yy}, α_{zz})	α_{iso}	E_{exc}
ADC(2)	(73.48, 38.69, 76.22)	62.80	(118.89, 26.65, 38.79)	61.45	4.32
ADC(2)-x	(77.21, 39.99, 79.97)	65.72	(114.28, 31.22, 51.10)	65.53	3.44
ADC(3/2)	(71.24, 38.27, 73.61)	61.04	(104.45, 33.64, 57.48)	65.19	4.50
CCSD Der.	(67.79, 37.50, 70.18)	58.49	(111.76, 42.13, 71.38)	75.09	4.59
CCSD E.V.	(70.05, 38.25, 72.52)	60.27	(102.93, 33.16, 51.14)	62.41	4.59

a) Data reproduced from Ref. 41.
b) Polarizability components (α_{AA}, α_{iso}) in au, excitation energies (E_{exc}) in eV.

For the pyrimidine molecule, a similar trend is observed for the agreement between computational results, which are shown in Table 7.2. Here, we find discrepancies $\delta_{\mathrm{Der.}}$ for ground state polarizabilities of 4.31, 7.23, 2.55, and 1.78 au for ADC(2), ADC(2)-x, ADC(3/2), and CCSD E.V., respectively. Deviations for the excited state (here $1^1\mathrm{B}_2$) to CCSD Der. are decreasing from ADC(2) (13.65 au), ADC(3/2) (9.90 au), to ADC(2)-x (9.56 au) and the difference to CCSD E.V. lies between ADC(3/2) and ADC(2) (12.68 au). Both excited states are of $n \rightarrow \pi^*$ character, and the largest increase in polarizability is found for in-plane components. This is consistent for all ADC methods in comparison to EOM-CCSD.[156] While all methods based on ISR/expectation values are capable of predicting these trends for *s*-tetrazine and pyrimidine correctly, one notices that trends for out-of-plane components (α_{zz} and α_{yy} for *s*-tetrazine and pyrimidine, respectively), are not in agreement with the EOM-CCSD derivative approach. However, this observation is made for ADC and CCSD E.V., thus, the effect is solely related to the ansatz to compute the polarizability, and not to the method itself. Note that coupling to the electronic ground state is negligible for the reported polarizabilities, since the $n \rightarrow \pi^*$ states are dipole-forbidden. Another observation that requires discussion is the reduced discrepancy to CCSD Der. results with increasing order of perturbation theory from ADC(2) toward ADC(3/2). In a recent study, it has been demonstrated that ADC(3/2) yields orbital relaxation effects for ph excited states through higher order of perturbation theory by including `pphh` states.[69] This explains the trends observed for the employed ADC schemes. Our numerical results thus yield the anticipated behavior of the respective ADC schemes, and provide values comparable to the related CCSD E.V. approach.

Formaldehyde and Naphthalene

Table 7.3: Static polarizabilities of the formaldehyde ground state and $1^1\mathrm{B}_1$.[a), b)]

State	ground state		$1^1\mathrm{B}_1$		
	$(\alpha_{xx}, \alpha_{yy}, \alpha_{zz})$	α_{iso}	$(\alpha_{xx}, \alpha_{yy}, \alpha_{zz})$	α_{iso}	E_{exc}
ADC(2)	(17.94, 12.88, 24.90)	18.57	(712.91, 243.79, 310.03)	422.25	6.27
ADC(2)-x	(18.40, 13.20, 25.65)	19.09	(641.26, 250.44, 314.84)	402.18	5.98
ADC(3/2)	(17.31, 12.68, 23.47)	17.82	(678.81, 281.39, 432.19)	464.13	7.57
CCSD Der.	(17.23, 12.50, 22.58)	17.44	(680.32, 272.78, 384.16)	445.75	7.05
CCSD E.V.	(17.39, 12.66, 22.98)	17.67	(688.15, 272.27, 388.23)	449.55	7.05

a) Data reproduced from Ref. 41.
b) Polarizability components (α_{AA}, α_{iso}) in au, excitation energies (E_{exc}) in eV.

Two more molecules for which experimental data for excited state polarizabilities are available are presented in the following. First, we examine formaldehyde, for which the computational results are displayed in Table 7.3. For all methods, a rather small ground state polarizability is found, which largely increases when the molecule is in the $1^1\mathrm{B}_1$ excited state. An approximately 20-fold increase in isotropic polarizability is present for all computational methods. This is consistent with the experiment, which reports an isotropic polarizability for the ground state as 18.9 au and that of $1^1\mathrm{B}_1$ as approximately 410 ± 180 au.[158,175] As such, all computational result are well within the range of the experimentally obtained values. The large increase in polarizability can be rationalized from the Rydberg-type excitation of the state at hand which possesses a large exciton size. This also explains why all components of the polarizability tensor are larger compared to the electronic ground state.

Next, consider the ground state and $1^1\mathrm{B}_{3\mathrm{u}}$ polarizability of naphthalene. The corresponding results from computations and experiment are shown in Table 7.4. The per-

Table 7.4: Static polarizabilities of the naphthalene ground and $1^1\mathrm{B}_{3\mathrm{u}}$ states.[a), b)]

State	ground state		$1^1\mathrm{B}_{3\mathrm{u}}$		
	$(\alpha_{xx},\ \alpha_{yy},\ \alpha_{zz})$	α_{iso}	$(\alpha_{xx},\ \alpha_{yy},\ \alpha_{zz})$	α_{iso}	E_{exc}
ADC(2)	(182.04, 133.68, 69.29)	128.34	(178.48, 73.79, 44.13)	98.80	4.45
ADC(2)-x	(194.22, 140.69, 71.34)	135.42	(164.97, 88.51, 50.91)	101.47	3.46
ADC(3/2)	(177.81, 129.95, 68.62)	125.46	(170.71, 98.76, 56.14)	108.53	4.16
CCSD Der.	(166.78, 123.14, 66.67)	118.86	(195.12, 121.47, 70.32)	128.97	4.41
CCSD E.V.	(173.89, 128.11, 68.22)	123.41	(164.30, 88.58, 54.89)	102.59	4.41
Experiment[c)]	(162.0, 119.5, 70.9)	117.4	(186.9, 120.1, 76.9)	128.0	4.02

a) Data reproduced from Ref. 41.

b) Polarizability components (α_{AA}, α_{iso}) in au, excitation energies (E_{exc}) in eV.

c) References 176 and 157.

centage deviations from experimental values are depicted in Figure 7.2. The performance of the computational methods compared to experiment is rather heterogeneous for the ground state polarizability. The largest overestimation for in-plane components α_{xx} and α_{yy} is found for ADC(2)-x with approximately 20%, whereas CCSD Der. agrees best with the experimental results for these components. Deviations from the experimental α_{zz} result are below 5%, except for CCSD Der. which underestimates the component by approximately 6%. As such, all employed methods except for ADC(2)-x yield reliable static polarizabilities for the electronic ground state of naphthalene. In the experiment, a small increase was observed for the static polarizability components of the $1^1\mathrm{B}_{3\mathrm{u}}$ state compared to the ground state, the largest of which is found for α_{xx}. For the polarizabilities based on expectation values, this trend could not be observed in the computational results. Especially the α_{yy} and α_{zz} components are largely underestimated by expectation-value-based methods, the most extreme being ADC(2) with more than -40% deviation for α_{zz}. On the contrary, derivative-based EOM-CCSD is capable of describing the trend of small increases in the components correctly. Here, the deviations are below 5% for α_{xx} and α_{yy}, and approximately -9% for α_{zz}. Thus, one can conclude that in this case amplitude relaxation effects seem to be especially important to model the polarizabilities correctly.

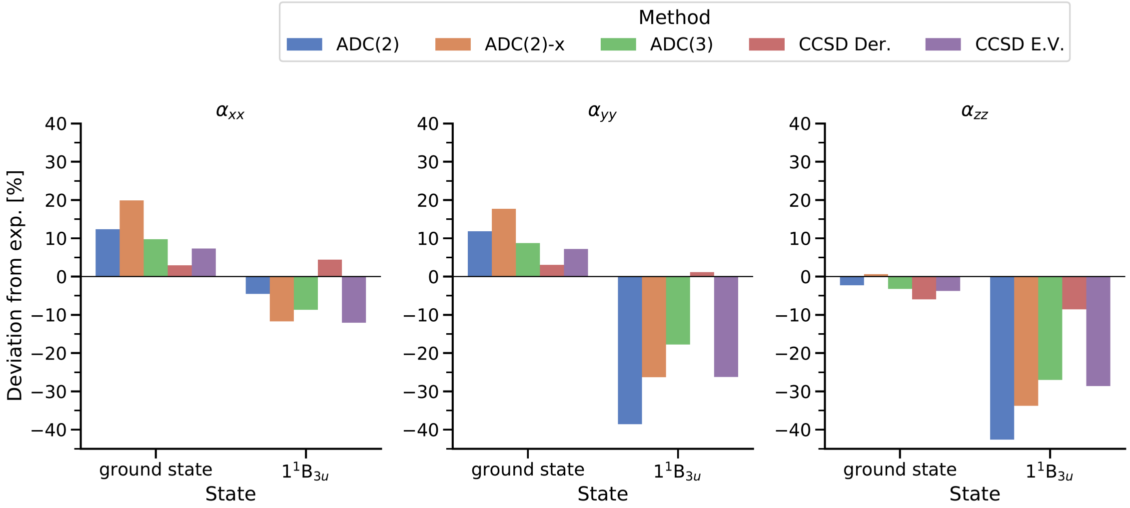


Figure 7.2: Deviations of computed polarizability components from the experimental value in percent for the ground and 1^1B_{3u} states of naphthalene.[1]

Uracil and *p*-Nitroaniline

As another example, the ground state and lowest singlet $n \to \pi^*$ ($1^1A''$) and $\pi \to \pi^*$ ($1^1A'$) states of uracil are considered, with results presented in Table 7.5. For the electronic ground state, all five methods again yield comparable results. Similar to *s*-tetrazine and pyrimidine, the $n \to \pi^*$ states have slightly increased polarizabilities for in-plane components α_{xx} and α_{yy}, when derivative-based EOM-CCSD is used. This is not the case for the expectation-value methods. In fact, all ADC and the EOM-CCSD E.V. results show a noticeable drop in polarizability for this state of uracil. Surprisingly, this discrepancy is not reduced when employing ADC(3/2), but instead becomes even larger. EOM-CCSD E.V. here yields values similar to ADC(3/2). For the $\pi \to \pi^*$ transition, both dipole moments and polarizabilities show a large increase.[156] For this state the α_{xx} component of the polarizability increases the most, due to the large coupling matrix element to the ground state. Again, ADC(3/2) and EOM-CCSD E.V. show a much smaller increase for this component than EOM-CCSD Der.

Results for *p*NA are shown in Table 7.6, including polarizability components of the electronic ground state and the lowest singlet excited $\pi \to \pi^*$ state (2^1A_1). Ground state polarizabilities are again similar. The probed singlet state corresponds to a strong intramolecular CT excitation.[156] As such, the corresponding dipole moment increases upon excitation, yielding a species with more ionic character than in the ground state. The excitation still shows a large oscillator strength, i.e., transition dipole moment along

Table 7.5: Static polarizabilities of the uracil ground state, $1^1\mathrm{A}''$, and $2^1\mathrm{A}'$.[a), b)]

State	ground state		$1^1\mathrm{A}''$			$2^1\mathrm{A}'$		
	(α_{xx}, α_{yy}, α_{zz})	α_{iso}	(α_{xx}, α_{yy}, α_{zz})	α_{iso}	E_{exc}	(α_{xx}, α_{yy}, α_{zz})	α_{iso}	E_{exc}
ADC(2)	(105.23, 80.00, 43.37)	76.20	(86.44, 67.30, 27.08)	60.28	4.73	(138.88, 81.77, 35.46)	85.37	5.32
ADC(2)-x	(110.75, 83.17, 44.48)	79.47	(85.17, 73.16, 33.17)	63.83	3.97	(147.08, 87.72, 38.37)	91.06	4.61
ADC(3/2)	(98.47, 76.27, 42.30)	72.35	(74.72, 70.74, 36.03)	60.50	5.36	(105.12, 86.93, 42.27)	78.11	5.38
CCSD Der.	(95.82, 74.62, 41.59)	70.67	(98.08, 88.29, 45.19)	77.19	5.22	(133.94, 102.78, 51.69)	96.14	5.58
CCSD E.V.	(98.75, 76.41, 42.42)	72.53	(71.49, 69.43, 34.55)	58.49	5.22	(112.82, 85.86, 42.97)	80.55	5.58

[a)] Data reproduced from Ref. 41.

[b)] Polarizability components (α_{AA}, α_{iso}) in au, excitation energies (E_{exc}) in eV.

the z-axis. Therefore, the polarizability of the $1^1\mathrm{A}_1$ largely increases in the α_{zz} component for all presented methods, particularly using CCSD Der. ADC(3/2) and EOM-CCSD E.V. behave similarly for the excited state, with a deviation $\delta_{\mathrm{Der.}}$ to EOM-CCSD Der. of 27.37 au and 29.04 au, respectively. Discrepancies of ADC(2) and ADC(2)-x are larger by approximately 10 au, amounting to 39.17 au and 40.91 au, respectively. Hence, the $\pi \rightarrow \pi^*$ intramolecular CT state shows the largest differences between derivative- and expectation-value-based methods studied here. The effects of full amplitude response in case of a CT excitation seem to have a large impact on the excited state polarizability of the respective state. As such, care should be taken in these cases. Nevertheless, ADC methods are capable to capture the trend of an increasing α_{zz} component for CT excitations correctly, whereas methods like time-dependent DFT tend to fail in this case.[177,178]

To summarize this brief study of static excited state polarizabilities, the presented findings both match the expected trends and previously published results, suggesting that the implementation is comparable to related methodologies using an expectation-value-based ansatz. The agreement between methods solely depends on the approach to evaluate the

Table 7.6: Static polarizabilities of the pNA ground state and $2^1\mathrm{A}_1$.[a), b)]

State	ground state		$2^1\mathrm{A}_1$		
	(α_{xx}, α_{yy}, α_{zz})	α_{iso}	(α_{xx}, α_{yy}, α_{zz})	α_{iso}	E_{exc}
ADC(2)	(118.84, 58.94, 168.68)	115.49	(68.26, 49.52, 196.97)	104.92	4.30
ADC(2)-x	(125.29, 60.62, 183.45)	123.12	(82.44, 50.60, 176.49)	103.18	3.56
ADC(3/2)	(112.76, 58.02, 162.94)	111.24	(86.53, 59.25, 204.37)	116.72	4.23
CCSD Der.	(106.38, 56.95, 152.90)	105.41	(109.60, 83.60, 239.08)	144.09	4.62
CCSD E.V.	(110.76, 58.20, 157.68)	108.88	(74.95, 69.30, 200.89)	115.05	4.62

a) Data reproduced from Ref. 41.

b) Polarizability components (α_{AA}, α_{iso}) in au, excitation energies (E_{exc}) in eV.

polarizability, and not whether ADC or CC is chosen. Hence, it would also be interesting to see how derivative-based ADC excited state polarizabilities would compare to derivative-based EOM-CCSD. In this case, amplitude-relaxed second derivatives of the ADC excited state energy would need to be derived and implemented. In addition, note that the ISR-based ansatz requires much less computational effort, yielding excited state polarizabilities for the price of ground state polarizabilities, once the excited states are determined. Using a derivative-based approach, however, more response equations need to be solved.[156]

Dispersion Coefficients for Excited States

Until now, only considered static polarizabilities of excited states were under examined, which do not require solutions of the complex response function. For C_6 dispersion coefficients, however, the isotropic average of the molecular dipole polarizability as a function of purely imaginary frequencies is needed to compute the interaction between two systems through the Casimir-Polder potential.[52,179] With the Python function to solve eq (7.5) in place, the required Gauss-Legendre integration can be easily carried out using built-in NumPy functions,[99] as shown in the code snippet in Figure 7.3. This code example shows again how well `adcc` integrates with the Python ecosystem, making it possible to quickly implement new features with only minor effort. In addition, the rich feature set of NumPy makes it possible to write code that strongly resembles the text book equations.

Using the code shown in Figure 7.3, C_6 dispersion coefficients for excited states of pyridine, pyrazine, and s-tetrazine were computed, previously studied with multi-configurational complete active space (CAS) calculations using a derivative-based approach.[166] The re-

```python
def c6_dispersion_coefficient_excited_state(state, state_of_interest):
    # set up points and weights
    points, weights = np.polynomial.legendre.leggauss(12)
    w0 = 0.3
    freqs = w0 * (1 - points) / (1 + points)
    alphas_iso = np.zeros_like(freqs)
    for i, w in enumerate(freqs):
        # compute polarizability
        re, im = complex_polarizability_excited_state(
            state, state_of_interest=state_of_interest, omega=0.0, gamma=w
        )
        # compute isotropic average
        alphas_iso[i] = 1.0 / 3.0 * (re[0, 0] + re[0, 3] + re[0, 5])
    derivative = w0 * 2 / (1 + points)**2
    integral = np.sum(alphas_iso * alphas_iso * weights * derivative)
    return 3.0 * integral / np.pi
```

Figure 7.3: Python function to compute the C_6 dispersion coefficients with `adcc/respondo`.[1]

sults for ADC(2), ADC(2)-x, ADC(3/2), and CAS are summarized in Table 7.7. The ground state C_6 dispersion coefficients for these molecules computed with ADC methods are all larger than the respective CAS results by approximately 18-30%. This deviation, however, becomes smaller from ADC(2) to ADC(3/2). In the case of excited state C_6 coefficients, one observes the opposite: all values obtained with ADC methods are smaller than the corresponding CAS value. In the case of pyrimidine, this amounts to dispersion coefficients that are 27%, 13%, and 2% smaller for ADC(2), ADC(2)-x, and ADC(3/2), respectively. This trend is even more pronounced for pyrazine, where decreases with respect to CAS results by 35%, 26%, and 12% for the ADC method hierarchy are found.

Table 7.7: C_6 dispersion coefficients of electronic ground and excited states employing ADC and CAS.[a)]

		C_6 dispersion coefficient [au]			
System	State	ADC(2)	ADC(2)-x	ADC(3)	CAS[b)]
pyridine	ground state	1717.23	1770.02	1682.04	1374
	1^1B_2	923.17	1100.90	1248.39	1278
pyrazine	ground state	1512.76	1559.15	1478.81	1245
	1^1B_{3u}	743.57	849.04	1007.38	1147
s-tetrazine	ground state	1161.05	1197.54	1129.03	919.6
	1^1B_{3u}	499.53	575.77	687.24	835.0

a) Data reproduced from Ref. 41.
b) Obtained from Ref. 166.

The findings are quite similar also for *s*-tetrazine with discrepancies of 40%, 31%, and 18%. These deviations to CAS results can be explained through the different methodologies used in the approaches. The C_6 coefficients are just derived properties of the excited state polarizabilities, which have been extensively discussed and compared previously. Hence, for the expectation-value-based ADC results, which only describe relaxation effects through the ADC matrix itself, it is expected to see differences compared to results from fully relaxed multi-configurational CAS. This behavior of the ADC hierarchy of methods is again corroborated by the fact that ADC(3/2) results, which include the most relaxation, deviate least from CAS values. Thus, the same behavior that was already observed for static excited state polarizabilities in previous calculations is observed also for complex frequency-dependent polarizabilities. The general formulation and implementation in `adcc` allows to evaluate the polarizabilities for open-shell molecules on top of an unrestricted Hartree–Fock reference, or for systems with few-reference character using the spin-flip ansatz.[180]

7.4 Conclusions and Outlook

In this chapter, the frontier of response functions within the ADC/ISR framework was pushed one step further to arbitrary response functions with the example of complex, frequency-dependent excited state polarizabilities. The combination of `adcc` and `respondo` is ideal to prototype novel properties and test them appropriately without compromise in performance. Since the building blocks for all response properties within ADC/ISR are virtually identical, and the ISR expressions can be directly obtained from the SOS form, it would be desirable to automate this implementation process. Such a library, named **`responsefun`**, using SymPy for symbolic algebra operations,[102] is currently under development by myself and my master student A. Papapostolou. The pilot implementation can be used with the following workflow, illustrated in Listing 7.1. First, the user conveniently specifies a response function by means of a symbolic SOS expression. The short-hand notation `TransitionMoment` creates a term of the type $\langle n|\hat{d}|f\rangle$. In the example, the SOS term for the complex ground state polarizability is given (see eq (2.78)). The variables `op_a` and `op_b` are symbolic representation of the electric dipole operator, whereas `O` and `n` are labels for the ground state and n-th excited state, respectively. Second, the symbolic frequencies are assigned an actual value. Third, the routine to evaluate ADC/ISR response properties `evaluate_property_isr` is called with the SOS expression, summation index, and real and complex frequencies as arguments. This function automatically follows the recipes shown in Chapter 2, i.e., it converts the SOS expression to ADC/ISR form, looks

```
alpha_terms = (
    + TransitionMoment(0, op_a, n) * TransitionMoment(n, op_b, 0) / (w_n - w -
    ↪  1j*gamma)
    + TransitionMoment(0, op_b, n) * TransitionMoment(n, op_a, 0) / (w_n + w +
    ↪  1j*gamma)
)
omega_alpha = [(w, 0.59)]
alpha_tens = evaluate_property_isr(
    state, alpha_terms, [n], omega_alpha, gamma_val=0.001
)
alpha_tens_sos = evaluate_property_sos(
    state, alpha_terms, [n], omega_alpha, gamma_val=0.001
)
```

Listing 7.1: Sample code for `responsefun` to compute the complex frequency-dependent ground state polarizability.

for special terms to be added, and collects response functions to be solved in a tree-like structure. Only unique matrix inversions are solved, which is rather tedious to implement manually otherwise. Finally, the function calls the `respondo` routine `solve_response` on each unique system of linear equations and assembles the output tensor automatically. The same procedure is also possible by direct evaluation of the SOS expression with the input `ExcitedStates` object named `state` in the example code. In the function `evaluate_property_sos`, the transition moments of all excited states are used to evaluate the response property tensor. When providing *all* ADC excitations to the function, it serves as a reference for the iteratively determined ADC/ISR value. Hence, the automated implementation provides direct access to reference data. Even though our implementation is at a very early stage, it will make manual implementations of ADC/ISR response properties obsolete in the future. One could then just use a textbook expression for the SOS of almost any response property and obtain the corresponding ADC result by conveniently typing in this equation into the `responsefun` interface. The `responsefun` library will be freely available as an add-on to `respondo` and `adcc` once the code is properly tested and the user interface is consolidated.

Notes

1. Reprinted from M. Scheurer, T. Fransson, P. Norman, A. Dreuw, and D. R. Rehn, “Complex Excited State Polarizabilities in the ADC/ISR Framework”, *J. Chem. Phys.* **2020**, *153*, 074112. With the permission of AIP Publishing.

Chapter 8

Distortion of Molecules Through External Forces

In this chapter, I investigate more extreme molecular environments. The first part presents a new electronic structure method for modeling molecules under high pressures in a black-box manner, which was derived and implemented from scratch.[181] Together with the derivation, first example applications to pressure-induced chemical reactions are presented. The novel method clearly outperforms previously existing approaches, especially with respect to versatility and ease of use. In the second part, the rupture mechanism of the metalloprotein, rubredoxin, is analyzed by means of a newly designed workflow, combining classical molecular dynamics (MD) simulations with state-of-the-art quantum chemical strain analysis tools. Through the hybrid workflow, the predominantly accepted mechanism for the extremely low rupture force of the protein's active site was questioned and shown to be most likely incomplete.[182] The complex environment of the iron-sulfur cluster in rubredoxin thus remains a challenge for future analyses.

Parts of this chapter have already been published in:

- M. Scheurer, A. Dreuw, E. Epifanovsky, M. Head-Gordon, and T. Stauch, "Modeling Molecules under Pressure with Gaussian Potentials", *J. Chem. Theory Comput.* **2021**, *17 (1)*, 583-597. (Reference 181)
- M. Scheurer, A. Dreuw, M. Head-Gordon, and T. Stauch, "The Rupture Mechanism of Rubredoxin is More Complex Than Previously Thought", *Chem. Sci.* **2020**, *11*, 6036-6044. (Reference 182)

8.1 Modeling Hydrostatic Pressure with the GOSTSHYP Approach

In the past decades, the interest in high-pressure chemistry has been remarkable,[183–185] both through the availability of various experimental techniques[186,187] and computational methods complementing the experimental findings.[188] The computational approaches are usually based on confining potentials using soft or hard boxes of varying shape and size,[189–198] and the agreement between computation and experiment is often remarkable.[199] These methods relying on confining potentials are hard to use in a black-box manner. Indeed, at the electronic structure level, no term in the molecular electronic Hamiltonian exists that would mediate an externally applied pressure. The most pragmatic approaches, which are universally applicable, have their origin in mechanochemistry,[188] where the pressure is "applied" via an external mechanical force to the nuclear energy gradient. An overview of such methods, e.g., the *Generalized Force-Modified Potential Energy Surface* (G-FMPES) approach,[200,201] the *Hydrostatic Compression Force Field* (HCFF)[202], or the recently published extended HCFF (X-HCFF) can be found in the literature.[203] As stated above, all mechanochemical pressure models mediate the external pressure through an additional term in the nuclear gradient, whereas the electron density, the SCF procedure, and the resulting molecular orbitals are blind for the pressurized environment. Another shortcoming is that mechanochemical models can only be applied to systems with at least two atoms. An alternative method for the application of pressure to chemical systems is the *eXtreme Pressure Polarizable Continuum Model* (XP-PCM).[204–206] Therein, an atom or a molecule is placed inside a cavity. Pressure is "applied" by reducing the cavity size and increasing the Pauli repulsion term of the solvent medium. The applied pressure is then calculated via a fitting procedure, *a posteriori*, as the negative partial derivative of the free energy with respect to the volume. Note that the user is here not capable of precisely defining the input pressure, and as such, no geometry optimizations at constant pressure are possible. Still, XP-PCM allows the modeling of chemical reactions,[205–207] spectroscopic properties,[208,209] as well as electronic[210] and structural[204,211,212] changes of chemical systems under pressure.

In this section, the development of a new electronic structure method for the simulation of atoms and molecules under pressure, the *Gaussians On Surface Tesserae Simulate HYdrostatic Pressure* (GOSTSHYP) approach, is introduced. The idea for the GOSTSHYP approach was conceived by T. Stauch and M. Head-Gordon, and in a collaborative project I was able to execute the first complete and correct derivation and implementation as explained in the following.[181] Similar to XP-PCM, GOSTSHYP considers a chemical system inside a cavity that is confined by a tessellated approximation of the van der Waals surface. Pressure is applied via a field of Gaussian potentials located at the tessellation points, confining the electron density. GOSTSHYP solves the shortcomings of previous methods for the simulation of chemical systems under pressure in that a) the method allows for geometry optimizations and AIMD simulations under user-defined pressures, b) both atoms and molecules can be subjected to pressure and c) the compression of electron density due to pressure is modeled realistically. GOSTSHYP is verified by comparing against a range of literature values for energies, geometries, dipole moments and chemical reactions under pressure. The GOSTSHYP method was implemented in Q-Chem and is available as of version 5.4.[94] At present, electronic structure calculations at the levels of Hartree-Fock[45,46] and density functional theory (DFT)[213,214] are supported. GOSTSHYP enables, by construction, the separate calculation of the pure influence of pressure on atoms or molecules, and therefore provides a unique and new point of view on pressure-induced chemical processes. In the following, the theoretical derivation of the GOSTSHYP model is presented.

8.1.1 Theoretical Background

In GOSTSHYP, hydrostatic pressure is modeled through a set of Gaussian potentials located at the molecule's van der Waals surface, which is discretized and tessellated through a Lebedev grid.[215] In this procedure, atom-centered spheres with the scaled (pressure-free) atom-specific van der Waals radii are superimposed and the overlapping regions are omitted, leading to a physically sound tessellation field. Since atomic van der Waals radii are in fact pressure-dependent,[216] in the following I refer to the scaling factors of the atom-centered spheres as the "tessellation sphere scaling factors" (TSSF). Each tessera j with the area A_j is assigned a Cartesian Gaussian function of the form

$$G_j(\mathbf{r}) = p_j \exp\left(-w_j(\mathbf{r} - \mathbf{r_j})^2\right) = p_j \tilde{G}_j(\mathbf{r}). \tag{8.1}$$

Here, p_j is the amplitude of the Gaussian centered at $\mathbf{r_j}$, whereas w_j is referred to as the width parameter. Using the electronic one-particle density matrix elements D_{pq} and the molecular orbitals $\{\phi_p\}$, the energy penalty by pressure is given by

$$E_\mathrm{p} = \sum_j \sum_{pq} D_{pq} \left\langle \phi_p | G_j(\mathbf{r}) | \phi_q \right\rangle . \tag{8.2}$$

This expression includes three-center overlap integrals, in which the mid center, i.e., the Gaussian, corresponds to an s-type function. In the following, determination of the function parameters w_j and p_j will be outlined. The simplest approximation is that the tessera areas are circular. Making this assumption and acknowledging that not all areas are equal (cf. Figure 8.1), the radius $\tilde{r}_j$ of the circle j that holds the Gaussian G_j can be calculated by

$$\tilde{r}_j = \sqrt{\frac{A_j}{\pi}}. \tag{8.3}$$

The goal of choosing an appropriate width parameter is to create a field of Gaussians that is as smooth and continuous as possible, because a continuous hydrostatic pressure should be modeled. Furthermore, "local high-pressure areas", where Gaussians of the same width and amplitude are densely packed, should be avoided (the green points in Figure 8.1).

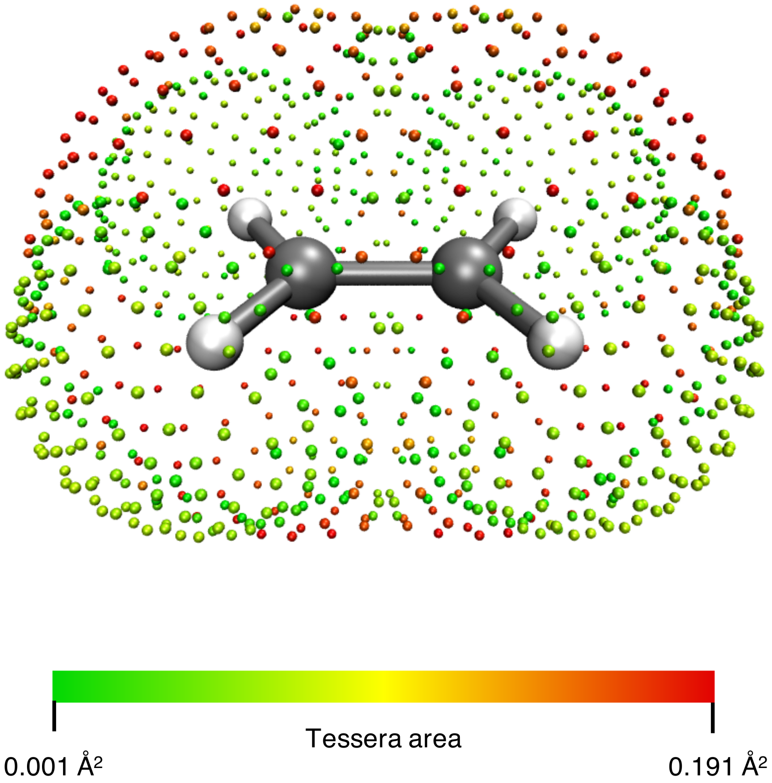


Figure 8.1: In the employed surface tessellation routine, some tesserae have larger areas than others. In this picture, red signifies large areas, whereas green represents small areas. Figure created by T. Stauch.[1]

Hence, it is reasonable to demand that at the edge of each tessera the amplitude of the accommodated Gaussian reaches half of its maximum value. This way, when neighboring Gaussians that belong to tesserae with similar areas overlap at the border between the two tesserae, their amplitudes add up to the value each of the Gaussian has in the middle of its tessera. Clearly, each Gaussian reaches its maximum when $\mathbf{r} = \mathbf{r}_j$, i.e., at the center of each tessera. At this point, the amplitude is $G_{j,\max} = p_j$. Requiring the value of w_j such that $G_{j,\frac{1}{2}\max} = \frac{1}{2}p_j$ at the border of the tessera leads to

$$w_j = \frac{\pi \ln 2}{A_j}. \tag{8.4}$$

The amplitudes p_j of the Gaussian functions modeling the pressure potential need to be defined such that the user-defined input pressure is exerted on the embedded molecule. At the interface between the molecule and the surrounding medium, the force acting from the outside ($\mathbf{F}_{\text{outer}}$), with which the field of Gaussian potentials compresses the electron density, needs to cancel the force acting from the inside ($\mathbf{F}_{\text{inner}}$), with which the electron density "pushes back". Hence, the requirement is

$$\mathbf{F}_{\text{outer}} = -\mathbf{F}_{\text{inner}}, \tag{8.5}$$

which implies that the absolute values of the force also need to match, i.e.,

$$|\mathbf{F}_{\text{outer}}| = |\mathbf{F}_{\text{inner}}| \tag{8.6}$$

or

$$F_{\text{outer}} = F_{\text{inner}}. \tag{8.7}$$

Using the definition of pressure,

$$P = \frac{F_\perp}{A}, \tag{8.8}$$

one realizes that the force acting from the outside is already the normal force $F_\perp$. Hence,

$$F_{\text{outer}} = P_{\text{inp}} A_j, \tag{8.9}$$

where P_{inp} is the pressure that the user inputs and A_j is the surface area of tessera *j*.

To understand the definition of the force acting from the inside, we imagine taking a Gaussian and displacing it along the surface normal vector. Pushing the Gaussian inwards, i.e. closer to the molecule, results in an increase in energy. Pulling it outwards, i.e., away from the molecule, results in a lowering of energy. Hence, the derivative of the electronic energy w.r.t. the position of the Gaussian equals the force with which the molecule pushes the surroundings back. To obtain the force perpendicular to the surface, the scalar product of the derivative vector and the normal vector of the surface $\mathbf{n} = (n_x, n_y, n_z)^T$ is used:

$$\begin{aligned} F_{\text{inner}} &= \frac{\partial E_{\text{p}}}{\partial x_j} n_x + \frac{\partial E_{\text{p}}}{\partial y_j} n_y + \frac{\partial E_{\text{p}}}{\partial z_j} n_z \\ &= \sum_{pq} D_{pq} \left(\frac{\partial}{\partial x_j} n_x \langle \phi_p | G_j | \phi_q \rangle + \frac{\partial}{\partial y_j} n_y \langle \phi_p | G_j | \phi_q \rangle + \frac{\partial}{\partial z_j} n_z \langle \phi_p | G_j | \phi_q \rangle \right). \end{aligned} \tag{8.10}$$

Inserting eqs 8.9 and 8.10 into eq 8.6 yields

$$\begin{aligned} P_{\text{inp}} A_j &= p_j \sum_{pq} D_{pq} \Big(n_x \frac{\partial}{\partial x_j} \langle \phi_p | \tilde{G}_j | \phi_q \rangle + n_y \frac{\partial}{\partial y_j} \langle \phi_p | \tilde{G}_j | \phi_q \rangle + n_z \frac{\partial}{\partial z_j} \langle \phi_p | \tilde{G}_j | \phi_q \rangle \Big) \\ &= -p_j \sum_{pq} D_{pq} \Big(n_x \langle \phi_p | 2w_j (x - x_j) \tilde{G}_j | \phi_q \rangle + n_y \langle \phi_p | 2w_j (y - y_j) \tilde{G}_j | \phi_q \rangle \\ &+ n_z \langle \phi_p | 2w_j (z - z_j) \tilde{G}_j | \phi_q \rangle \Big). \end{aligned} \tag{8.11}$$

This expression includes three-center overlap integrals in which the mid center is a p-type function. Rearranging for p_j, the amplitude of Gaussian j, yields

$$p_j = -\frac{P_{\text{inp}} A_j}{\widetilde{F}_j}, \tag{8.12}$$

where

$$\widetilde{F}_j = \sum_{pq} D_{pq} \mathcal{F}_{j,pq} \tag{8.13}$$

and

$$\begin{aligned} \mathcal{F}_{j,pq} = & \, n_x \langle \phi_p | 2w_j (x - x_j) \tilde{G}_j | \phi_q \rangle \\ & + n_y \langle \phi_p | 2w_j (y - y_j) \tilde{G}_j | \phi_q \rangle \\ & + n_z \langle \phi_p | 2w_j (z - z_j) \tilde{G}_j | \phi_q \rangle . \end{aligned} \tag{8.14}$$

With the previous expressions at hand, the hydrostatic pressure energy for Gaussian j is given by

$$\begin{aligned} E_{\mathrm{p},j} &= \sum_{pq} D_{pq} \langle \phi_p | G_j | \phi_q \rangle \\ &= -\sum_{pq} D_{pq} \langle \phi_p | \frac{P_{\mathrm{inp}} A_j}{\widetilde{F}_j} \tilde{G}_j | \phi_q \rangle \\ &= -P_{\mathrm{inp}} A_j \sum_{pq} D_{pq} \langle \phi_p | \tilde{G}_j | \phi_q \rangle \left(\sum_{pq} D_{pq} \mathcal{F}_{j,pq} \right)^{-1} . \end{aligned} \tag{8.15}$$

The total contribution of the set of Gaussians to the energy, $\mathrm{E_p}$, can be calculated as

$$E_{\mathrm{p}} = \sum_j E_{\mathrm{p},j} \ , \tag{8.16}$$

which entails the computation of three-center overlap integrals. The required integral routines were implemented based on the Obara-Saika scheme[217] in the Q-Chem 5.4 program package[93,94].

SCF calculations of atoms and molecules in the GOSTSHYP scheme require the contribution to the Fock operator due to the presence of the Gaussian potentials, which reads

$$\begin{aligned} V_{j,pq} = \frac{\partial E_{\mathrm{p},j}}{\partial D_{pq}} &= -P_{\mathrm{inp}} A_j \frac{\partial}{\partial D_{pq}} \sum_{rs} D_{rs} \langle \phi_r | \tilde{G}_j | \phi_s \rangle \left(\sum_{rs} D_{rs} \mathcal{F}_{j,rs} \right)^{-1} \\ &= \langle \phi_p | G_j | \phi_q \rangle + P_{\mathrm{inp}} A_j \left[\sum_{rs} D_{rs} \langle \phi_r | \tilde{G}_j | \phi_s \rangle \right] \mathcal{F}_{j,pq} \left(\sum_{rs} D_{rs} \mathcal{F}_{j,rs} \right)^{-2} . \end{aligned} \tag{8.17}$$

Hence, the Fock operator contribution through the Gaussian potentials is updated in each SCF iteration with the current one-electron density matrix. It is not a “simple” operator contribution to the core Hamiltonian, but introduces a non-linearity due to its density dependence. This structure is similar to polarizable environment models like PE, however, the GOSTSHYP operator terms require more bookkeeping and non-standard building blocks in their implementation. Since the current one-electron density matrix is used for setting up the Gaussian potentials in each SCF step and the tessellation field is re-calculated in each step of a geometry optimization or AIMD simulation, the pressure selected by the user, P_{inp}, is applied throughout the entire course of the calculations. For geometry optimizations and AIMD simulations, analytical nuclear gradients were derived and implemented. The explicit derivation is shown in the following.

Analytic Nuclear Gradients A short-hand notation for integrals according to

$$\langle \phi_p | G_j | \phi_q \rangle = \langle p | G_j | q \rangle \tag{8.18}$$

will be used. Recall the definition of the Fock operator (eq (2.36)) in the presence of the Gaussian potentials,

$$\begin{aligned} f_{pq} = h_{pq} &+ \sum_{rs} D_{rs} \left(\langle pr|qs \rangle - \langle pr|sq \rangle \right) \\ &+ \sum_j \left[\langle p|G_j|q \rangle - P_{\text{inp}} A_j \left[\sum_{rs} D_{rs} \langle r|\tilde{G}_j|s \rangle \mathcal{F}_{j,pq} \left(\sum_{rs} D_{rs} \mathcal{F}_{j,rs} \right)^{-2} \right] \right] . \end{aligned} \tag{8.19}$$

Let

$$E_{\text{SCF}} = \sum_{pq} D_{pq} h_{pq} + \frac{1}{2} \sum_{pqrs} D_{pq} D_{rs} \left(\langle pr|qs \rangle - \langle pr|sq \rangle \right) + V_{\text{nn}} + \sum_j \sum_{pq} D_{pq} \langle p|G_j|q \rangle \tag{8.20}$$

be the total SCF energy in presence of Gaussian potentials. Now consider the derivative of the SCF energy with respect to a general perturbation λ,[47,218]

$$\begin{aligned} \frac{\partial E_{\text{SCF}}}{\partial \lambda} = &\sum_{pq} \frac{\partial D_{pq}}{\partial \lambda} h_{pq} + \sum_{pq} D_{pq} \frac{\partial h_{pq}}{\partial \lambda} + \frac{1}{2} \sum_{pqrs} \left(\frac{\partial D_{pq}}{\partial \lambda} D_{rs} + D_{pq} \frac{\partial D_{rs}}{\partial \lambda} \right) \left(\langle pr|qs \rangle - \langle pr|sq \rangle \right) \\ &+ \frac{1}{2} \sum_{pqrs} D_{pq} D_{rs} \frac{\partial}{\partial \lambda} \left(\langle pr|qs \rangle - \langle pr|sq \rangle \right) + \frac{\partial V_{\text{nn}}}{\partial \lambda} \\ &+ \sum_j \frac{\partial}{\partial \lambda} P_{\text{inp}} A_j \sum_{pq} D_{pq} \langle p|\tilde{G}_j|q \rangle \left(\sum_{pq} D_{pq} \mathcal{F}_{j,pq} \right)^{-1} . \end{aligned} \tag{8.21}$$

Everything but the last term is identical to the vacuum analytical SCF energy gradient, so only the final term is now considered separately for each tessellation point j, i.e.,

$$E^{\lambda}_{\mathrm{p},j} = \frac{\partial}{\partial\lambda} P_{\mathrm{inp}} A_j \sum_{pq} D_{pq} \langle p|\tilde{G}_j|q\rangle \left(\sum_{pq} D_{pq}\mathcal{F}_{j,pq}\right)^{-1} \tag{8.22}$$

$$\begin{aligned} &= \underbrace{P_{\mathrm{inp}} \frac{\partial A_j}{\partial\lambda} \sum_{pq} D_{pq} \langle p|\tilde{G}_j|q\rangle \left(\sum_{pq} D_{pq}\mathcal{F}_{j,pq}\right)^{-1}}_{E^{\lambda,(1)}_{\mathrm{p},j}} \\ &\quad \underbrace{+ P_{\mathrm{inp}} A_j \left(\sum_{pq} D_{pq}\mathcal{F}_{j,pq}\right)^{-1} \left(\frac{\partial}{\partial\lambda} \sum_{pq} D_{pq} \langle p|\tilde{G}_j|q\rangle\right)}_{E^{\lambda,(2)}_{\mathrm{p},j}} \\ &\quad \underbrace{+ P_{\mathrm{inp}} A_j \sum_{pq} D_{pq} \langle p|\tilde{G}_j|q\rangle \frac{\partial}{\partial\lambda} \left(\sum_{pq} D_{pq}\mathcal{F}_{j,pq}\right)^{-1}}_{E^{\lambda,(3)}_{\mathrm{p},j}} . \end{aligned} \tag{8.23}$$

The first term in eq (8.23) contains the cavity derivative,[219,220] which can be computed with the present routines for the non-electrostatic PCM nuclear gradients. Carrying out the derivatives in the second term, $E^{\lambda,(2)}_{\mathrm{p},j}$, yields

$$E^{\lambda,(2)}_{\mathrm{p},j} = p_j \sum_{pq} D_{pq} \frac{\partial \langle p|\tilde{G}_j|q\rangle}{\partial\lambda} \tag{8.24}$$

$$= p_j \sum_{pq} D_{pq} \left(\langle \frac{\partial p}{\partial\lambda} |\tilde{G}_j|q\rangle + \langle p|\tilde{G}_j|\frac{\partial q}{\partial\lambda}\rangle + \langle p|\frac{\partial \tilde{G}_j}{\partial\lambda}|q\rangle \right) . \tag{8.25}$$

Note that all derivatives of the one-particle density matrix are already consumed into the energy-weighted density matrix contracted with the derivatives of the overlap matrix,[218] such that we only need to consider the derivative of the operator explicitly. Hence, the last term of eq (8.25) requires further attention,

$$\langle p|\frac{\partial \tilde{G}_j}{\partial\lambda}|q\rangle = -\frac{\partial w_j}{\partial\lambda} \langle p|(\mathbf{r}-\mathbf{r}_j)^2\tilde{G}_j|q\rangle - w_j \langle p|\frac{\partial}{\partial\lambda}(\mathbf{r}-\mathbf{r}_j)^2\tilde{G}_j|q\rangle , \tag{8.26}$$

where the first term contains a derivative of the width parameter w_j multiplied with a three-center overlap integral with a d-type function on the mid center, and the second term represents the Hellmann-Feynman force through the Gaussian potential. The derivative of the width parameter is straightforward, since it depends on the cavity derivative given by

$$\begin{aligned}\frac{\partial w_j}{\partial \lambda} &= \frac{\partial}{\partial \lambda}\frac{\pi \ln 2}{A_j} \\ &= \pi \ln 2 \frac{\partial}{\partial \lambda}\frac{1}{A_j} \\ &= -\pi \ln 2 \frac{\partial A_j}{\partial \lambda}\frac{1}{A_j^2}.\end{aligned} \tag{8.27}$$

Considering the last term of eq (8.23), $E_{\mathrm{p},j}^{\lambda,(3)}$, one obtains

$$E_{\mathrm{p},j}^{\lambda,(3)} = -P_{\mathrm{inp}} A_j \left(\sum_{rs} D_{rs} \langle r|\tilde{G}_j|s\rangle\right)\left(\sum_{rs} D_{rs}\mathcal{F}_{j,rs}\right)^{-2} \sum_{pq} D_{pq}\frac{\partial}{\partial \lambda}\mathcal{F}_{j,pq}. \tag{8.28}$$

The derivative of $\mathcal{F}_{j,pq}$ is expanded to

$$\begin{aligned}\frac{\partial}{\partial \lambda}\mathcal{F}_{j,pq} = \frac{\partial}{\partial \lambda}\Big(& n_x \langle p|2w_j(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle \\ & + n_y \langle p|2w_j(y-y_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle \\ & + n_z \langle p|2w_j(z-z_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle \Big).\end{aligned} \tag{8.29}$$

The first term (the two other terms can be treated analogously) yields three different contributions through the derivative, i.e.,

$$\begin{aligned}\frac{\partial}{\partial \lambda}\langle p|2w_j(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle = n_x\Big[& \langle \frac{\partial p}{\partial \lambda}|2w_j(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle \\ & + \langle p|2w_j(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|\frac{\partial q}{\partial \lambda}\rangle \\ & + \langle p|\frac{\partial}{\partial \lambda}2w_j(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle\Big],\end{aligned} \tag{8.30}$$

where the first two terms are simple derivatives of a three-center overlap integral on the “bra” and “ket” side, respectively. The last term is more involved due to the non-vanishing derivative of the width parameter,

$$
\begin{aligned}
&\langle p|\frac{\partial}{\partial\lambda}2w_j(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle \\
=\;&2\frac{\partial w_j}{\partial\lambda}\langle p|(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle \\
&+2w_j\langle p|\frac{\partial}{\partial\lambda}(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle \\
=\;&2\frac{\partial w_j}{\partial\lambda}\langle p|(x-x_j)\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right)|q\rangle \\
&+2w_j\langle p|\exp\left(-w_j(\mathbf{r}-\mathbf{r}_j)^2\right) \\
&\times\left[\frac{\partial(x-x_j)}{\partial\lambda}-2(x-x_j)(\mathbf{r}-\mathbf{r}_j)\frac{\partial(\mathbf{r}-\mathbf{r}_j)}{\partial\lambda}w_j-(x-x_j)(\mathbf{r}-\mathbf{r}_j)^2\frac{\partial w_j}{\partial\lambda}\right]|q\rangle\,. \qquad (8.31)
\end{aligned}
$$

The red terms in the above equations arise from the standard derivative of the Gaussian with perturbation-independent exponents, whereas the blue term gives rise to an overlap integral involving (Cartesian) f-type orbitals on the mid center multiplied with the derivative of the width parameter given in eq (8.27). Collecting and assembling the terms for the other two Cartesian components is straightforward.

8.1.2 Computational Methodology

All calculations were run using a development version of the Q-Chem 5.3 program package[93,94] in which the GOSTSHYP approach was first implemented. I implemented the required three-center overlap integrals using an Obara-Saika scheme[217] for arbitrary angular momentum and first analytical nuclear derivatives in Q-Chem's `libqints` library. Unit tests against reference results generated with Mathematica[221] were implemented. The analytical derivatives were verified through comparison with finite difference result using a five-point stencil. The GOSTSHYP model is implemented in the new `distort` library of Q-Chem, where I also added other pressure models with a consistent and maintainable interface. The `distort` library has its own section in the Q-Chem input file through which the pressure model and all other parameters can be selected. An example input file for the geometry optimization of a water molecule at 50 GPa with GOSTSHYP is shown in Listing 8.1. The `distort` library is maintained and updated by the group of T. Stauch. For the calculations atomic van der Waals radii by Bondi were used.[222] Various levels of theory, i.e., Hartree-Fock[45,46] and DFT[213,214] with different basis sets, were used in the benchmark and example computations, the details can be found in the published paper.[181] Atomic and molecular volumes were estimated using a numerical integration scheme.[223] Cube files for the ground state electron density were generated in Q-Chem

```
$molecule
0  1
H   -3.5008791    1.2736107    0.7596000
O   -3.9840791    1.3301107   -0.0574000
H   -4.9109791    1.2967107    0.1521000
$end

$rem
jobtype         opt
method          hf
basis           vdz
use_libqints    1
gen_scfman      1
distort         1
$end

$distort
 model              gostshyp
 pressure           50000
 npoints_heavy      302
 npoints_hydrogen   302
$end
```

Listing 8.1: Q-Chem example input file for the geometry optimization of water with GOSTSHYP. With the `$distort` section in the input, all model parameters can be set by the user.

using the standard boundaries of the mesh box (± 3Å around the maximum/minimum of the molecular coordinates) and a grid spacing of 0.1 Å. The number of grid points with an electron density larger than 0.001 a.u.[223] was divided by the number of total grid points and multiplied with the box volume yielding the molecular volume. The code to estimate the molecular volume based on "cube files" is shown in Listing 8.2. The integration scheme was compared to results from Ref. 223 and was found to agree well with their Monte-Carlo technique (data not shown). Born-Oppenheimer *ab initio* molecular dynamics (BOMD) simulations of the Diels-Alder reaction between cyclopentadiene and ethylene were run using an integration time step of 20 au. Ten independent simulations were run for 1800 time steps at a temperature of 298 K. Initial velocities were randomly generated from a Maxwell-Boltzmann distribution using the given temperature.

```python
import numpy as np
from scipy import constants
import sys

outfile = sys.argv[1]
plotfile = sys.argv[2]
box = None
with open(outfile, "r") as of:
    ls = of.readlines()
    ls = [l.strip() for l in ls]
for i, l in enumerate(ls):
    if "Grid information in Angstroms:" in l:
        gridinfo = ls[i+1:i+4]
        box = []
        for g in gridinfo:
            sp = [col for col in g.split(" ") if col != ""]
            box.append(float(sp[7])-float(sp[4]))

a0 = constants.value("Bohr radius") * 1e10
thresh = 0.001  # density isovalue threshold
x, y, z = box
v_box = x * y * z
cube = np.loadtxt(plotfile, skiprows=3)
density = cube[:, 3]
count = np.count_nonzero(density > thresh)
# MC integration
V = v_box * count / density.size
V_au = V / (a0**3)
```

Listing 8.2: Python code to compute molecular volumes using Monte-Carlo integration of the electron density cube file.

8.1.3 Results and Discussion

GOSTSHYP Benchmarks

As a first step, the GOSTSHYP method was tested for reliable and consistent behavior when changing the adjustable parameters. The number of tessellation points per atom N_{points} and the tessellation sphere scaling factors (TSSF) determine a) the density of the Gaussian potentials and b) the “distance” of the potential field to the molecule itself. We chose to benchmark these parameters against the related XP-PCM[204–206] with the example of *trans*-1,3-butadiene.[204] First, the carbon-carbon double bond length under pressure with a fixed TSSF of 1.2 and varying N_{points} was evaluated.[181] The analysis showed a converged bond length already at $N_{\text{points}} = 110$. Only minor changes in bond length were observed when varying the TSSF parameter, such that the typical value for PCM of 1.2 was chosen as the default. One could also see that GOSTSHYP produced very smooth pressure-dependent geometrical properties of *trans*-1,3-butadiene, in contrast to XP-PCM.[181] Note that it becomes mandatory to increase the TSSF for computations on multiple molecular fragments to avoid generation of tessellation points between the individual fragments. From the parameter study, we concluded that GOSTSHYP can be used in a black-box manner using the above default values without having to determine the optimal parameters for each system under study.

The main motivation for GOSTSHYP was the possibility to compute pressure-dependent electronic energies of atoms and molecules. To this end, the electronic energies of the hydrogen atom and the H_2^+ cation under pressure were calculated with GOSTSHYP and compared to exact analytical values from literature.[181] These reference values were obtained by placing the hydrogen atom inside differently sized boxes with impenetrable walls.[195] The agreement between GOSTSHYP and the literature values of the hydrogen electronic energy under pressure was quite remarkable up to a pressure of approximately 3000 GPa.[181] Going beyond several TPa of hydrostatic pressure is, from a practical point of view, not required.[186] This analysis demonstrates the striking difference between GOSTSHYP and existing mechanochemical pressure models,[200,202,203] which only mediate the external pressure through a nuclear gradient contribution. I want to point out that my implementation directly worked for unrestricted open-shell references due to the fact that the Gaussian potentials are of course spin-independent and only “see” the entire electron density of the system.

For the other open-shell test case, the H_2^+ cation, the results obtained from GOSTSHYP computations agree well with literature values. These reference values were produced by placing the ion inside a spheroidal box.[192] The benchmarks for hydrogen atoms and the H_2^+ cation showed that pressure-dependent electronic energies with the GOSTSHYP approach are chemically sound and in good agreement with established pressure models. Clearly, GOSTSHYP provides a great advantage over purely mechanochemical pressure models through its integration of the pressure energy penalty in the molecular Hamiltonian. The compression of the electron density was further quantified by estimation of molecular volumes, as explained above.[181] Further benchmark samples and analyses are to be found in the GOSTSHYP publication,[181] including the dipole moment of water under pressure. Now, we turn our attention to the practical application of GOSTSHYP in investigating pressure-induced chemical reactions in a static and dynamic manner.

Pressure-Induced Chemical Reactions

The first example for a pressure-induced reaction performed with GOSTSHYP was the Diels-Alder reaction of cyclopentadiene and ethylene.[184,224] This reaction has already been investigated using XP-PCM,[206] where a pressure of 11.2 GPa at zero temperature was enough to induce the cycloaddition. To model the same reaction with GOSTSHYP, a TSSF of 1.8 was used to avoid the generation of tessellation points between the two fragments. The application of pressure to a complex of cyclopetadiene (diene) and ethylene (dienophile) via the static GOSTSHYP approach (T = 0K) leads to a monotonic decrease in the distance d between the diene and the dienophile (Fig. 8.2). Initially, the distance shows two small downward steps, due to to minor changes in relative orientation and conformation. Between 39 and 40 GPa, a sudden downward jump of d indicates the formation of the product through a pressure-induced Diels-Alder reaction. From 40 GPa onward, no pronounced compression of the adduct is observed. The pressure required to induce the Diels-Alder reaction with GOSTSHYP is approx. 3.5 times higher than in XP-PCM,[181] however, both models are fundamentally different in their way of determining the applied pressure. Note that with XP-PCM, no geometry optimizations under pressure are possible, indicating a clear advantage of GOSTSHYP since analysis of pressure-dependent potential energy surfaces is the cornerstone for reliable modeling of chemical reactions. Going one step further, it is also possible to carry out *ab initio* molecular dynamics (AIMD) simulations using GOSTSHYP. To showcase these capabilities, proof-of-principle Born-Oppenheimer molecular dynamics (BOMD) simulations of cyclopentadiene and ethylene under a constant pressure of 40 GPa at 298 K were carried out. Throughout the ten simulated trajectories, the average distance between the carbon atoms in cyclopentadiene and ethylene that form bonds during the Diels-Alder reaction oscillates (Fig. 8.3), which is a result of the compressing influence of the Gaussian potentials and the repulsion of the electron density of the molecules. During the simulation time, in nine out of ten trajectories, a Diels-Alder cyclization was observed, which is indicated by a steep decrease of the average distance between cyclopentadiene and ethylene during bond formation. After carbon-carbon bonds have been formed, the average distance still oscillates, which is again a result of the interplay between compression due to the Gaussian potentials and the restoring force of the molecule. These oscillations occur with an increased frequency as soon as the bonds are formed, which is due to the larger force constant of the covalent bonds compared to the intermolecular non-covalent coordinate. The oscillation amplitude decreases over time due to equilibration of the reaction product.[181] Clearly, the applicability of GOSTSHYP in AIMD simulations is an outstanding and important fea-

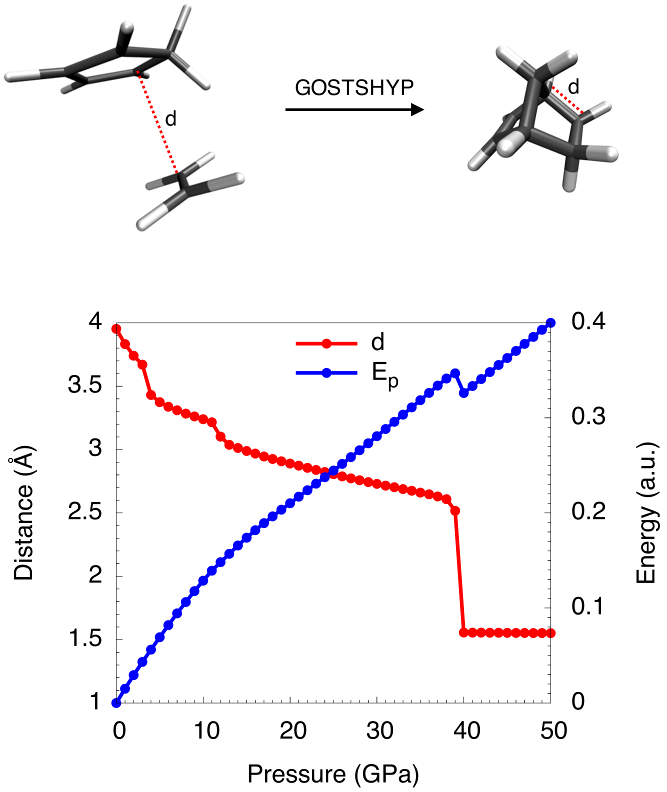


Figure 8.2: Pressure-induced Diels-Alder reaction of cyclopentadiene and ethylene, modeled with GOSTSHYP at the PBE[225]/cc-pVDZ[128] level of theory. d, which refers to the distance between the carbon atoms in cyclopentadiene and ethylene that form a bond during this reaction, is plotted as a function of pressure. The contribution of the pressure to the energy (E_p) is plotted as well. Figure produced by T. Stauch.[1]

ture, compared to related methods, since it allows investigation of the interplay between pressure and temperature in a time-resolved manner on the single molecule level.

Another illustrative example of the capabilities of GOSTSHYP in the simulation of pressure-induced chemical reactions is the cyclotrimerization of acetylene under pressure yielding benzene.[226,227] Three acetylene molecules were placed in the same plane to mimic the configuration on the surface of a heterogeneous catalyst. Geometry optimizations of this arrangement of molecules at different pressures were carried out at the B3LYP[228–230]-D3BJ[231]/6-31G(d)[143] level of theory. The optimization results for a pressure range from 0 to 100 GPa are shown in Figure 8.4. Indeed, the pressure exerted on the individual acetylene molecules through GOSTSHYP ultimately leads to the formation of benzene at a pressure of 68 GPa. The average bond length of newly formed carbon-carbon bonds (cf. Fig. 8.4) steadily decreases up to 67 GPa. At 68 GPa and higher pressures, the carbon-

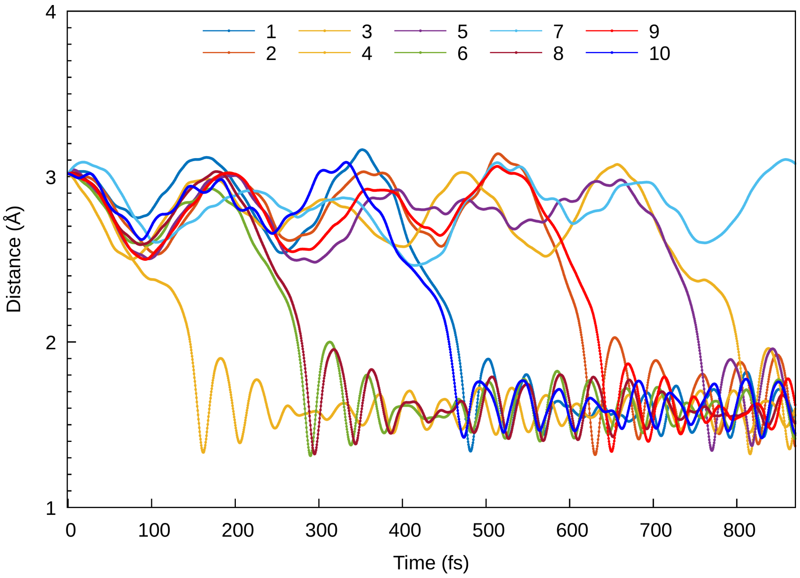


Figure 8.3: Time-dependent progression of the average distance between the carbon atoms in cyclopentadiene and ethylene that form a bond during the Diels-Alder reaction during the BOMD simulation under a constant pressure of 40 GPa, applied via the GOSTSHYP model at the PBE[225]/cc-pVDZ[128] level of theory. Each line represents an individual trajectory.[1]

carbon bond lengths remain at an almost constant value of 1.37 Å. Such high pressures are impossible to reach experimentally in the gas phase, where a catalyst is used to obtain benzene out of acetylene.[226]

8.1.4 Conclusions and Outlook

In this section, an electronic structure method for geometry optimizations and AIMD simulations of chemical systems under pressure, the *Gaussians On Surface Tesserae Simulate HYdrostatic Pressure* (GOSTSHYP) approach, was introduced. Ground state SCF energies, together with analytic nuclear gradients were derived and implemented. The user-defined input pressure specified for GOSTSHYP is applied during geometry optimizations and AIMD simulations, and the compression of electron density is described realistically since the pressure dependence is directly integrated into the molecular Hamiltonian through a density-dependent energy penalty. Benchmarks against established pressure models and exact analytical results were successful, and GOSTSHYP is also capable of modeling the pressure-dependent Diels-Alder reaction of cyclopentadiene and ethylene as well as the cyclotrimerization of acetylene. GOSTSHYP is a black-box method for both experts and novices in the field of modeling pressure-dependent quantities, and

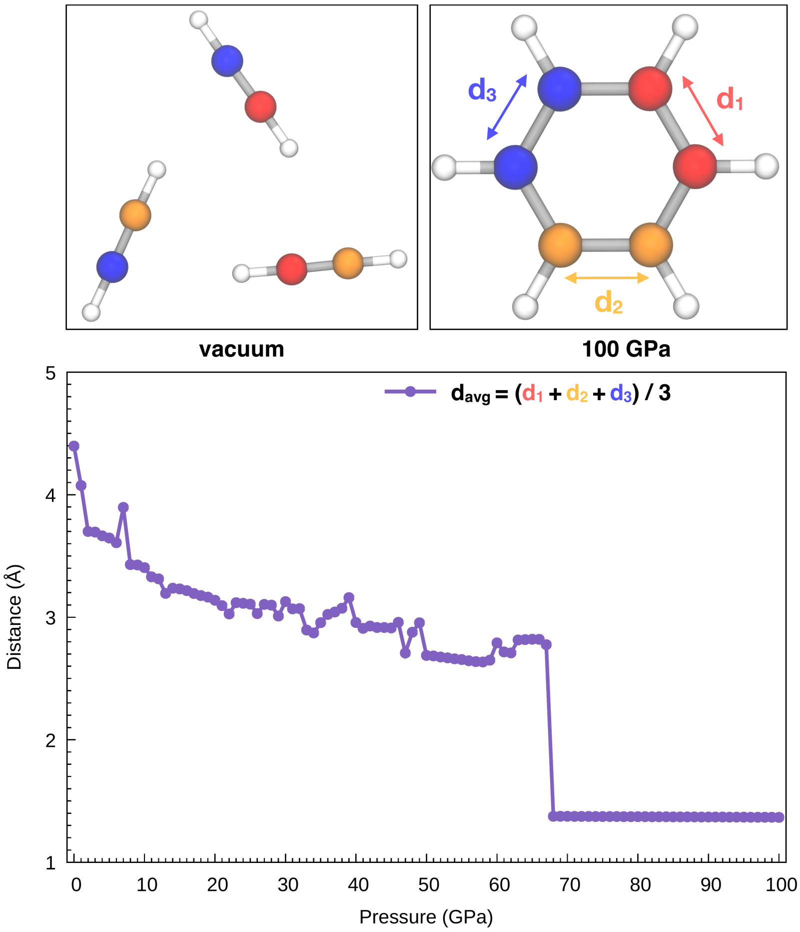


Figure 8.4: Pressure-induced cyclotrimerization of three acetylene molecules to benzene. The upper panels show the vacuum-optimized geometry with three distinct acetylene molecules and the GOSTSHYP geometry at a pressure of 100 GPa. Carbon atoms involved in newly formed bonds are equally colored. The lower panel shows the average bond distances d_{avg} of the newly formed bonds at different pressures.[1]

the method was implemented in the Q-Chem program package at the SCF level, enabling Hartree-Fock and DFT calculations. It is one of the "highlighted" new features of the recent Q-Chem 5.4 release through the newly implemented `libdistort` library (`https://www.q-chem.com/explore/pes/distort`).

An interesting extension of the here presented approach would be the computation of electronically excited states under pressure, e.g., with TDDFT or ADC. This would enable the simulation of UV/Vis spectra of compressed molecules.[232–235] A quick test of GOSTSHYP together with ADC (data not shown) indicated that maybe the combination is not as straightforward and must be more thoroughly devised in future work. For improved performance, it would be beneficial to screen the required three-center overlap integrals in general and maybe to derive a simplified GOSTSHYP model that requires less gradient terms. The most cumbersome terms arise from the exponents of the Gaussian potential functions that depend on the molecular geometry through the surface areas, i.e., if one could find a way to fix the exponent of all potentials in a reasonable manner, the computational procedure of GOSTSHYP would simplify a lot. Another approach would be the evaluation of the Gaussian potentials via grid integration techniques: one could project the (adaptive) Gaussian potentials on a DFT integration grid and then evaluate the Fock operator contribution and energy via efficient DFT routines. This suggestion, however, is complicated by the fact that the potentials are dependent on the surface tessellation in a non-linear manner through the width parameter. Both proposed extensions and adaptations could in principle be quickly tested with the current implementation. The latter approach can also be piloted in the Psi4 project, which would make the GOSTSHYP method openly available to a broader user base, if successful.

8.2 The Rupture Mechanism of Rubredoxin

Proteins are constantly exposed to mechanical forces, from synthesis, folding, re-folding, unfolding and degradation.[236] The field of mechanobiology has been under extensive investigation throughout the past decades, using experimental and computational methodologies, or combined approaches.[237–239] In fact, proteins display a rich and diverse behavior when exposed to mechanical forces,[236,239] which can in some cases even be counterintuitive: Metalloproteins, for example, are often much weaker than one would expect from the covalent character of the involved metal-ligand bonds.[240,241] An intriguing example for this phenomenon is found in rubredoxin, a protein that participates in electron transfer reactions[242–244] and provides structural stability due to its central FeS_4 unit.[244] Using atomic force microscopy (AFM), it was found that the strong covalent character of the iron-sulfur bonds[245] of the protein's central FeS_4 unit contrasts strikingly with a remarkably low rupture force of 200 pN.[246] On the contrary, covalent bonds typically display rupture forces above 1.5 nN.[247] As iron-sulfur clusters are ubiquitous in biochemical systems, the mechanical properties of the rubredoxin active site were thoroughly investigated, both experimentally and computationally, in the past years. It was found that the cleavage of the Fe−S bond in rubredoxin typically proceeds via a homolytic pathway,[248,249] however, heterolytic bond rupture was described if nucleophiles are present.[250] Most important to the hypothetical mechanism devised in these seminal works is the environment of the central FeS_4 unit in rubredoxin,[250] and the mechanism and its kinetics of rupture depend on the pulling direction.[251] The mechanical anisotropy of rubredoxin has been confirmed through computational investigation by means of steered molecular dynamics (SMD) simulations.[252]

The presence of hydrogen bonds ($NH\cdots S$) between neighboring amino acid backbones and the sulfur atoms of the FeS_4 unit (Figure 8.5) was suggested as a possible explanation for the surprisingly low rupture force of the Fe−S bonds in rubredoxin[254] by decreasing the covalent character of the Fe−S bonds. Similar effects have been previously identified in the context of a modulation of reduction potentials in rubredoxin by hydrogen bonds.[255] Interestingly, however, rubredoxin model systems with intramolecular $NH\cdots S$ hydrogen bonds exhibit mean Fe−S bond lengths that are significantly shorter than those of comparable complexes that do not form hydrogen bonds.[256] Considering that a short bond is typically interpreted as strong,[237] this finding seems to be in conflict with the notion that $NH\cdots S$ hydrogen bonds weaken the Fe−S bonds in rubredoxin.

Motivated by these contradictions and the excellent review,[240] I collaborated with Prof. Dr. Tim Stauch to thoroughly investigate the hypothesized rupture mechanism of

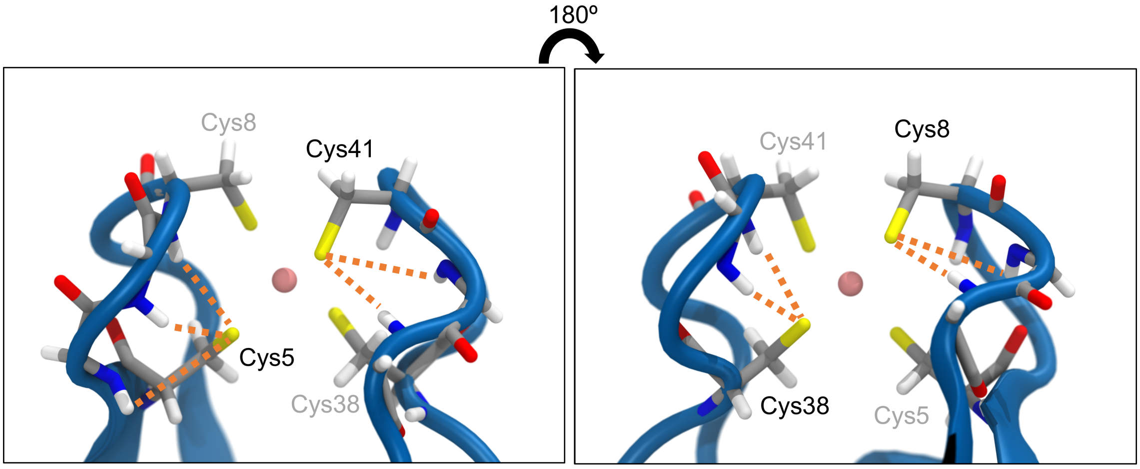


Figure 8.5: Close-up view of the central FeS_4 unit in rubredoxin (PDB 1BRF). Dotted lines represent possible hydrogen bonds between the sulfur atoms and the closest amino acid backbones (Cys5: Lys6, Ile7, Cys8; Cys8: Gly9, Tyr10; Cys38: Ile40, Cys41; Cys41: Gly42, Ala43) according to Ref. 253. The structure in the right panel is rotated by 180° with respect to the structure in the left panel.[2]

rubredoxin. Computational methodologies used up to this point were able to capture a lot of the experimentally observed properties, however, detailed analysis of hydrogen bond networks and most importantly, a quantum chemical investigation of the rupture process in the protein environment was lacking. To this end, we sought to combine SMD with quantum chemical strain analysis to describe the dynamic fluctuations of strain energy during forced unfolding. To come up with a suitable approach was quite difficult from a technical point of view. After several failed attempts to use QM/MM optimized input structures for strain analysis, we came up with an SMD-JEDI analysis[237] workflow that resolves the quantum mechanochemical properties with unprecedented spatial and temporal resolution. Furthermore, DFT is used to investigate the mechanical resilience and anisotropy of rubredoxin in detail. Hydrogen bonds between neighboring amino acid backbones and the sulfur atom of the central FeS_4 pseudo-tetrahedron are shown to play only a minor role in the mechanical properties of rubredoxin. The protein's mechanical resistance is influenced by structural anisotropy and angle bendings in the FeS_4 unit, as evidenced by state-of-the-art strain analyses. These effects are not sufficient to explain the experimentally observed low rupture force of rubredoxin, thus hinting at a rupture mechanism that is likely more complex than previously thought.

8.2.1 Computational Methodology

Atomic coordinates were obtained from a *Pyrococcus Furiosus* rubredoxin crystal structure (PDB: 1BRF).[257] Hydrogen atoms were added using the VMD[150] *psfgen* plugin, and the protein was placed inside a water box with 0.15 M NaCl. For all molecular dynamics (MD) simulations, the CHARMM36 force field[258] was applied together with NAMD, version 2.13.[259] The charges of Fe(III) and cysteine residues of rubredoxin were re-parametrized.[182] The anharmonic Fe-S bond potential was modeled through a Morse potential with parameters taken from previously published work ($D_e = 90$ kJ/mol, $\beta = 30$ nm^{-1}, $r_0 = 2.3$ Å).[252] The system was first minimized and subsequently equilibrated at a temperature of 300 K with a time step of 2 fs for 1 ns in total. During equilibration, backbone atoms of the protein were harmonically constrained with a force constant of $k = 1.0\ \mathrm{kcal \cdot mol^{-1} \cdot Å^{-2}}$. Afterwards, a longer equilibration simulation was carried out for 25 ns without any constraints. For further analysis of hydrogen bonds to cysteine residues in the active site, an extended equilibration simulation of 250 ns length was performed. Starting from the equilibrated protein structure, SMD simulations were carried out to unfold the rubredoxin protein. In these SMD simulations, the Cα atom of Ala1 were fixed in space, whereas the Cα atom of the C-terminal Asp53 was pulled away from Ala1 along the bond axis at a constant velocity of 0.2 nm/ns employing a force constant of $k = 7.0\ \mathrm{kcal \cdot mol^{-1} \cdot Å^{-2}}$. Snapshots of the trajectory were saved every 20 ps and the SMD trajectories were run until at least the first Fe$-$S bond rupture occurred. Ten SMD runs were run in total, and for the strain analysis a single trajectory was considered in the following due to the enormous computational cost of the workflow. Hydrogen bond analysis in equilibrium and in all ten SMD trajectories was carried out using PyContact,[260] taking into account distance criteria from Ref. 253 (S$-$H distance ≤ 3.0 Å). Possible hydrogen bond donors are Lys6, Ile7, Cys8, Gly9, Tyr10, Ile40, Cys41, Gly42, and Ala43.[253] Distance analyses were performed with MDAnalysis.[261,262]

To obtain an improved temporal resolution of the bond rupture event, the chosen SMD trajectory was restarted 1 ns before bond rupture while saving a snapshot every 0.1 ps. The quantum chemical Judgement of Energy DIstribution (JEDI) analysis was applied to investigate the partitioning of strain energy among all bonds, bendings and torsions of a mechanically strained molecule.[263–265] Although force analyses by molecular dynamics simulations have been conducted for proteins,[266,267] this is the first study to carry out the quantum chemical JEDI analysis along the unfolding trajectory of a protein. The workflow for this mixed quantum mechanical/molecular mechanics (QM/MM) approach is illustrated in Figure 8.6. Snapshots including the four cysteine residues and Fe(III)

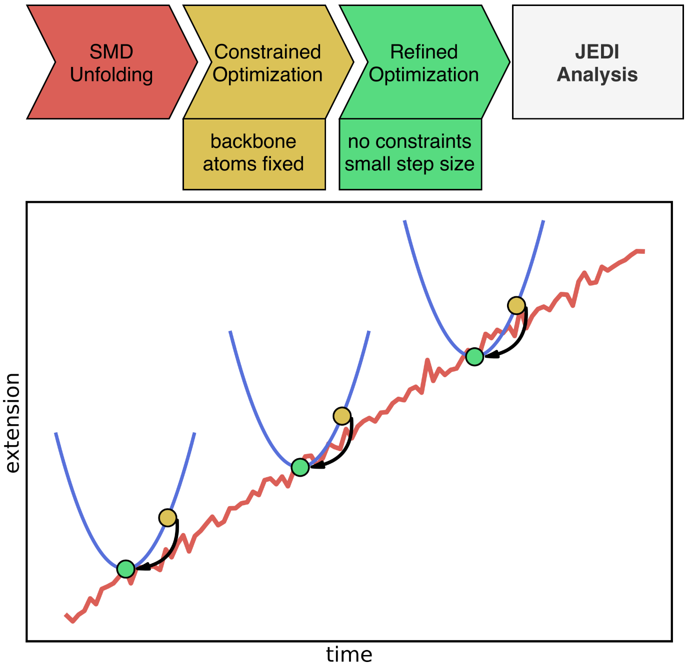


Figure 8.6: Schematic illustration of the applied workflow for JEDI analysis along the SMD unfolding trajectory. The upper part shows the individual steps toward JEDI analysis. The (non-quantitative) plot illustrates the SMD trajectory leading to a certain extension of the protein (red). Certain snapshots are extracted and quantum chemically optimized keeping protein backbone atoms fixed (yellow dots). Then, the nearest local minimum, needed for JEDI analysis, is found through a constraint-free optimization of the previous geometry (green dot). In this manner, pairs of strained and relaxed structures along the unfolding trajectory are obtained, yielding a well-defined JEDI analysis per snapshot.[2]

were extracted from the previous trajectory up to 9 ps before bond rupture. Broken bonds were saturated with hydrogen atoms. For each snapshot, a geometry optimization with DFT[213,214] at the BP86VWN[228,268]/6-31G(d)[143] level of theory was performed while keeping the backbone atoms of the amino acids fixed to provide a strained geometry for each snapshot. The BP86VWN functional was chosen because it offers an attractive compromise between agreement with experimental data and computational effort. Afterwards, a follow-up optimization without constraints was carried out with a tiny step size (Q-Chem keyword `geom_opt_dmax=20`) to find the nearest local minimum, resulting in the relaxed reference geometry for that particular snapshot. Finally, JEDI analyses were carried out for each snapshot. The above protocol was implemented and carried out by myself. The computational methodology for static quantum chemical calculations on a minimal $[Fe(III)(SCH_3)_4]^-$ system,[248] performed by T. Stauch, is reported in detail in our joint publication.[182] The bonding of the FeS_4 cluster was analyzed by means of Localized Orbital Bonding Analysis (LOBA),[269] and the covalent character of the Fe−S bond was measured using the ALMO-EDA[270] charge transfer energy ΔE_{CT}. All static quantum chemical calculations were carried out using the Q-Chem 5.1 program package.[93,94]

8.2.2 Results and Discussion

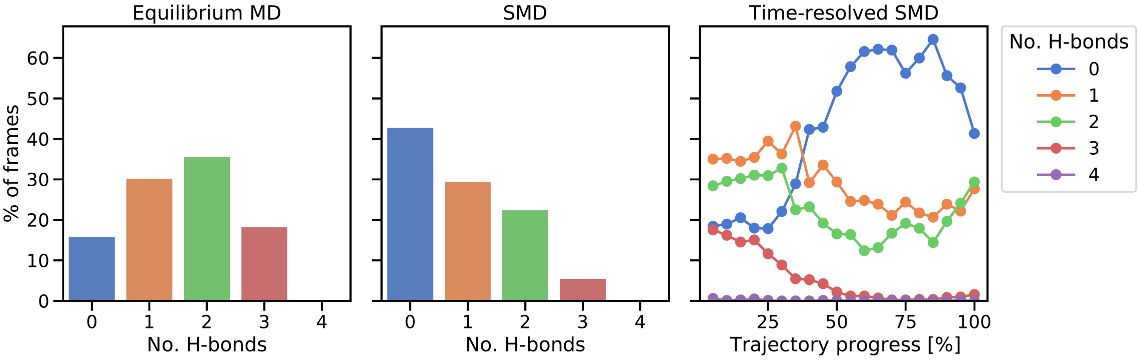


Figure 8.7: Analysis of protein hydrogen bonds to sulfur atoms of Cys5 and Cys41. The percentage of frames in which a certain number of hydrogen bonds occurs in equilibrium (left) and during the SMD unfolding procedure (middle). In the right panel, a time-resolved representation of hydrogen bond percentages is shown by averaging 20 equally sized trajectory windows. During the final parts of the trajectories, some of the Fe−S distances are already significantly elongated.[2]

In all SMD trajectories, one of the Fe−S bonds ruptures. In six of the ten trajectories, the sulfur atom involved in the bond rupture belongs to Cys5 and in the other four trajectories it belongs to Cys41. Hence, in a first step, the hydrogen bond network of the sulfur atoms belonging to Cys5 and Cys41 was analyzed in the equilibrium and in the steered unfolding trajectories (Figure 8.7) to determine the influence of the hydrogen bond network on the mechanical properties of rubredoxin. In the equilibrium trajectory, most frames display either one or two hydrogen bonds in Cys5 and Cys41, with a minority of frames showing either zero or three hydrogen bonds. While it is of course possible that these hydrogen bonds influence the reactivity of the central FeS_4 unit in the force-free state of rubredoxin, a similar analysis carried out during the unfolding trajectories prior to rupture of the Fe−S bond shows that the formation of NH···S hydrogen bonds becomes less and less frequent with increasing stress: In almost half of the frames during the unfolding trajectories, no such hydrogen bond is formed. The formation of one or two hydrogen bonds occurs less frequently and only a very small minority of frames throughout the unfolding events displays three hydrogen bonds. The prevalence of hydrogen bonds during the unfolding trajectory decreases almost continuously as rupture is approached (Figure 8.7, right panel). As an exception, in the last 5-10% of the shown trajectories hydrogen bonds on average become more frequent again. At these points, the mean iron-sulfur distances have already increased significantly and approach the value of a ruptured bond (see Fig. 8.8). In agreement with chemical intuition, the sulfur atoms tend to form

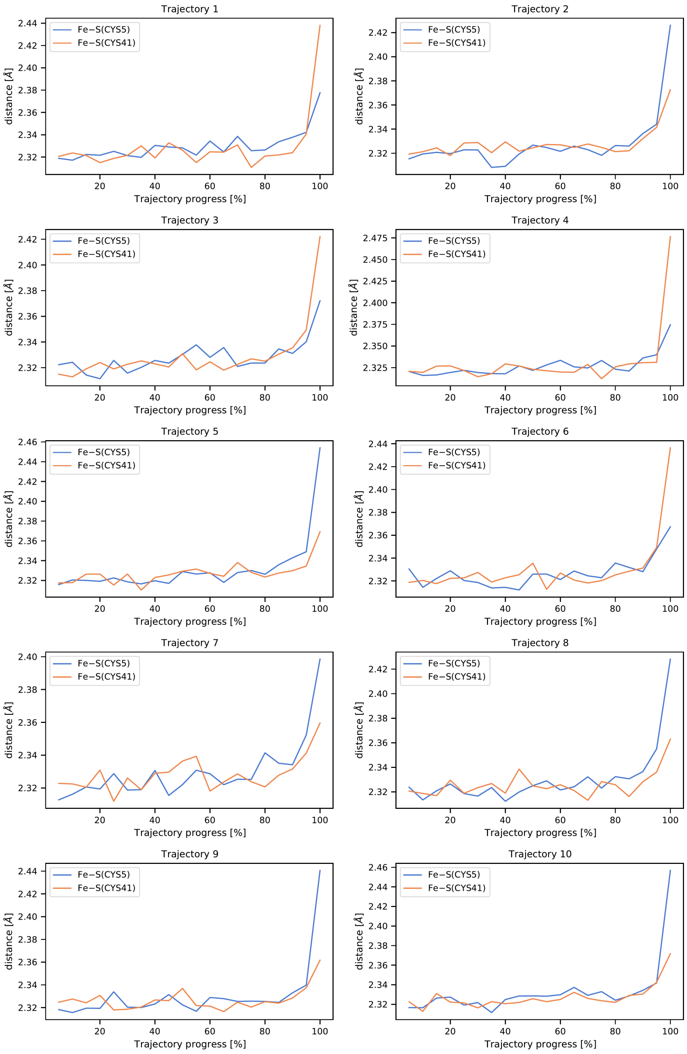


Figure 8.8: Fe−S distances averaged for 20 equally sized trajectory windows of all ten SMD trajectories. At 100% trajectory progress, the first bond ruptures.

stabilizing hydrogen bonds if this is sterically feasible, i.e., if they are not screened by the rest of the FeS_4 unit. Hence, up to the point of rupture of the iron-sulfur bond, the mean amount of hydrogen bonds to the sulfur atoms decreases. All in all, these observations demonstrate that hydrogen bonds to the sulfur atoms of the FeS_4 unit are unlikely to determine the low rupture force of rubredoxin found in the experiments, since close to the bond rupture of the Fe−S bond these hydrogen bonds become less prevalent.

Subsequently, we focused on elucidating the real-time propagation of strain energy in the central part of rubredoxin during unfolding. To this end, we applied the workflow summarized in Figure 8.6 to one of the SMD trajectories to derive strain energies using the JEDI analysis. As mentioned before, it is a stochastic process whether Cys5 or Cys41 ruptures. In the case of the particular unfolding trajectory considered here, the scissile bond is $Fe{-}S_5$ (where “5” represents amino acid residue Cys5), as can be observed from the progression of the Fe−S distances in Figure 8.9. Together with the $Fe{-}S_{41}$ bond, which is also being elongated to a certain extent, the scissile $Fe{-}S_5$ bond is part of the force-bearing scaffold of rubredoxin, because it lies along the connection line between the attachment point and the pulling point in the SMD simulation. The $Fe{-}S_5$ bond distance oscillates tremendously during the trajectory, which is a result of the dynamic nature of the calculations, and partially breaks and reforms several times during the last 9 ps before it is broken completely. The remaining Fe−S bonds ($Fe{-}S_8$ and $Fe{-}S_{38}$), in turn, stay very close to their equilibrium bond lengths, since the force is acting almost perpendicular to them.

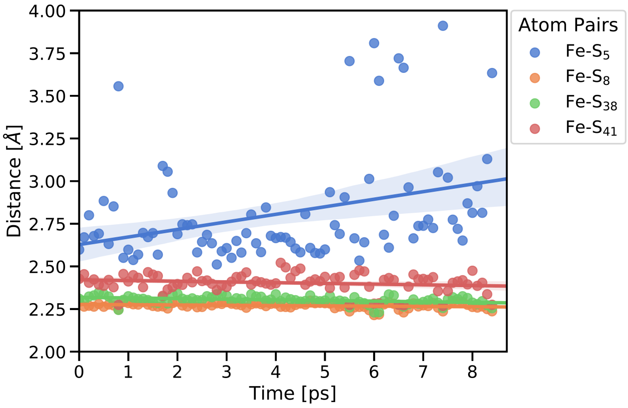


Figure 8.9: Fe−S distances of strained geometries before bond rupture. Subscripts indicate the amino acid residue number. Linear fits of the distances are displayed together with the confidence interval (translucent areas).[2]

To study the dynamic propagation of mechanical strain energy during the unfolding process, the JEDI analysis was carried out at each snapshot. Considering the last 9 ps before the ultimate scission of the $Fe{-}S_5$ bond, most strain energy is stored in the $Fe{-}S_5$ bond itself (Figure 8.10a). A significant amount of strain energy is also stored in the $Fe{-}S_{41}$ bond as well as in several bond angles that lie along the connecting line of the attachment point and the pulling point of the SMD simulation. Together, these internal coordinates comprise the force-bearing scaffold of the central part of rubredoxin. Dihedral angle displacements play only a minor role in this trajectory. The significant role of the Fe$-$S$_5$ bond and the Fe$-$S$_{41}-$C$_{41}$ bond angle as reservoirs of strain energy is further emphasized when considering the dynamic progression of strain energy (Figure 8.10b). As expected, the amount of strain energy in the scissile Fe$-$S$_5$ bond increases dramatically when approaching the point of bond rupture.

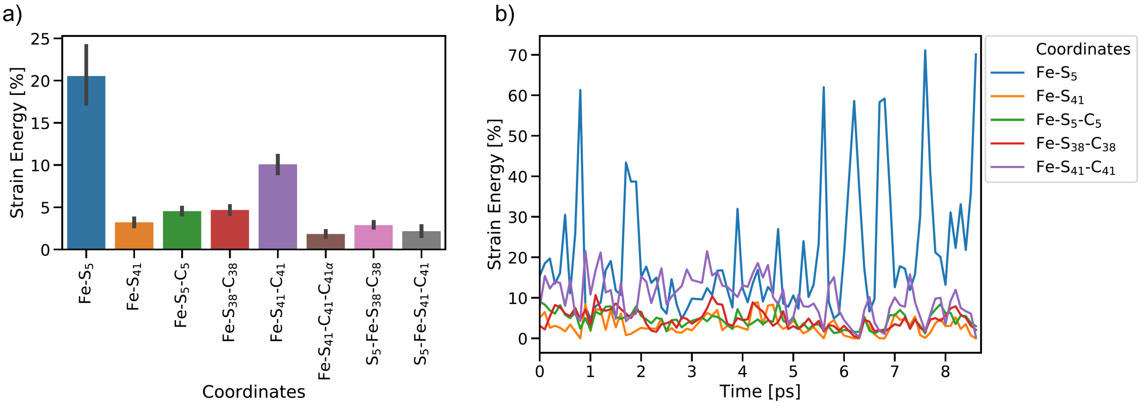


Figure 8.10: Summary of JEDI analyses including a) mean strain energy percentages for the most strained coordinates (bond lengths Fe$-$S$_5$ and Fe$-$S$_{41}$, bond angles Fe$-$S$_5-$C$_5$, Fe$-$S$_{38}-$C$_{38}$ and Fe$-$S$_{41}-$C$_{41}$, and torsions Fe$-$S$_{41}-$C$_{41}-$C$_{41\alpha}$, S$_5-$Fe$-$S$_{38}-$C$_{38}$ and S$_5-$Fe$-$S$_{41}-$C$_{41}$) and b) time-resolved strain energy contributions (percentages of the total strain) of the most important bond lengths and bendings. Bars around the mean percentages indicate the 95% confidence interval.[2]

The distribution of strain energy in a representative snapshot is summarized by using a color-coded representation in Figure 8.11. The force-bearing scaffold can be clearly distinguished: As expected, it lies along the connecting line between the attachment point and the pulling point. In the SMD trajectory, most strain energy is stored in the scissile Fe$-$S$_5$ bond, but the other side of the FeS$_4$ pseudo-tetrahedron is also strained significantly.

The results obtained with the new SMD-JEDI combination were then complemented with quantum chemical computations, carried out by my collaborator, to model the rupture process in detail in isolated model systems [Fe(III)(SCH$_3$)$_4$]$^-$ and [Fe(II)(SCH$_3$)$_4$]$^{2-}$.[182] For the following summary, I will limit myself to the Fe(III) system. The computed rupture force of the Fe(III) minimal model system was found to be 1.89 nN, which is much higher than the experimentally observed value in the protein system (258 ± 122 pN).[246] Even though this is a typical observation,[237,247,271] the drastic overestimation of the rupture force clearly indicated that the model system does not capture the experimental conditions,

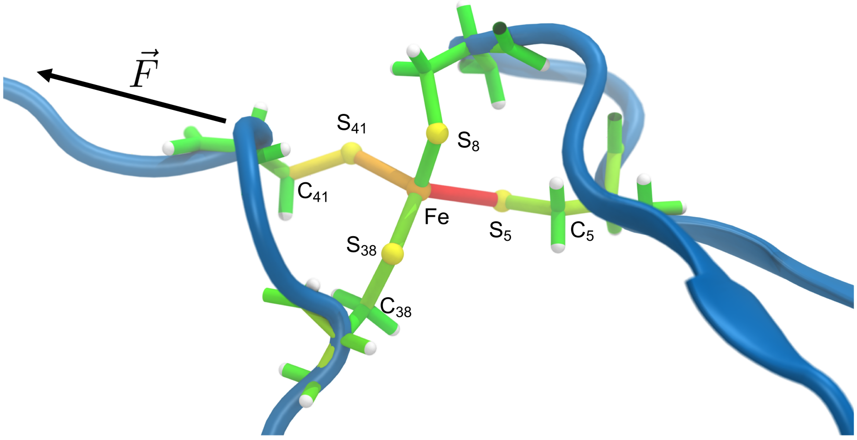


Figure 8.11: Representative snapshot of the unfolding trajectory before bond rupture. The strain energy contributions in the quantum region are mapped onto the bonds, where red and green indicate high and low strain, respectively, and transitions are fluent. The residues along the force coordinate bear most of the strain energy (Cys5 and Cys41), whereas Cys8 and Cys38 only play a minor role. The scissile bond Fe$-$S$_5$ contains most of the strain energy, followed by the angle on the opposite side (Fe$-$S$_{41}$$-C_{41}$).[2]

since such a high rupture force usually indicates covalent character of the scissile bond.[272] Using LOBA, the Fe(III)$-$S bond was characterized as covalent. To mimic the hydrogen bond network provided by rubredoxin, the hypothetical "key player" in the rupture mechanism, rupture forces were determined for the model system with formamide molecules in the environment. In the previous part, hydrogen bonds were shown to only play a minor role by means of MD simulations, however, we aimed at ruling out the possibility that hydrogen bonds can indeed dramatically reduce the computed rupture force. Formamide is a typical protein backbone mimicking molecule, and up to six formamide molecules were placed in proximity to the sulfur atoms of the model systems, forming hydrogen bonds like in the rubredoxin active site. These six hydrogen bonds were also observable in the SMD simulations and previous work.[254] Intriguingly, there is no consistent trend of the rupture force when adding more and more hydrogen bonds.[182] Instead, the rupture force is decreased by less than 10% with one or two hydrogen bonds, but increases again when adding two more hydrogen bonds. We concluded that the electronic environment of the iron-sulfur complex has a pivotal influence on the rupture force, but not in a manner that one would have expected from experiment. The oscillations of the model system's rupture force clearly show that hydrogen bonds with the protein backbone cannot explain the surprisingly low rupture force of rubredoxin. Using the JEDI[263–265] analysis, it was found that the distribution of strain energy in the complex is indeed anisotropic.[182] Most of the strain energy is stored in the scissile Fe−S bond, more than in the chemically equivalent adjacent Fe−S bond. A large portion of strain energy is furthermore distributed over apparently uninvolved parts of the pseudo-tetrahedron. This is consistent with the JEDI analysis carried out in the hybrid SMD-JEDI workflow. From all results presented herein, it can be concluded that the hydrogen bond network in rubredoxin is most likely not responsible for the extremely low rupture force.

8.2.3 Conclusions and Outlook

In this section, I presented the first combination of SMD and the quantum chemical JEDI strain analysis to show that hydrogen bonds from neighboring amino acid backbones to the sulfur atoms of the FeS_4 unit of rubredoxin become less prevalent when unfolding the protein and that they do not lower the rupture force.[182] This clearly precludes the assumption that hydrogen bonds cause the low rupture force found in the experiment. The protocol allowed us to track the distribution of strain energy during the mechanical unfolding of rubredoxin, providing a time-resolved view of the propagation of strain in the stretched protein in unprecedented detail. Hence, the SMD-JEDI protocol provides a significant step toward modeling complex biomolecular systems under mechanical strain with active sites that absolutely require quantum chemical accuracy and predictive power. It would be interesting to extend the here presented SMD-JEDI protocol to a hybrid QM/MM SMD approach. The prerequisites for this to be successful are, however, tightly converged structures and an analytic hybrid QM/MM Hessian taking into account all terms arising through the quantum-classical coupling scheme. This might provide an additional improvement over the here presented SMD-JEDI protocol, which can be viewed as a type of mechanical embedding scheme, whereas electrostatic embedding would model the interaction of the quantum region with point charges in the environment. The software packages used in this study are in principle capable of providing the required data, however, the systems need to be prepared and propagated very cautiously. Instead of the so far established hydrogen bond explanation, unobserved mechanisms most possibly prevail, which will need to be investigated in detail using new experiments and computations. The novel combination of SMD and quantum chemical strain analysis will prove as a valuable tool for this purpose, not only for rubredoxin but also for other biomolecular systems exposed to external forces. The study together with the new workflow is a stepping stone for future modeling of protein mechanochemistry and will pave the way for a rigorous understanding of such systems at quantum chemical detail.

Notes

1. Reprinted with permission from: M. Scheurer, A. Dreuw, E. Epifanovsky, M. Head-Gordon, and T. Stauch, "Modeling Molecules under Pressure with Gaussian Potentials", *J. Chem. Theory Comput.* **2021**, *17 (1)*, 583-597. Copyright 2021 American Chemical Society.

2. Reproduced from Ref. M. Scheurer, A. Dreuw, M. Head-Gordon, and T. Stauch, "The Rupture Mechanism of Rubredoxin is More Complex Than Previously Thought", *Chem. Sci.* **2020**, *11*, 6036-6044. With permission from the Royal Society of Chemistry.

Chapter 9

Design and Investigation of Novel Photocages

Photolabile protecting groups (PPGs), also called photocages, are molecules with the ability to release a biologically or chemically active substance upon light irradiation.[274,275] Photocages enable the controlled activation of a specific molecule with high spatial and temporal resolution through the underlying light-induced uncaging reaction. An efficient and practically usable photocage has to meet a plethora of requirements.[274] First, to avoid cellular damage in *in vivo* experiments, the photoreaction should be preferably triggered through light with wavelengths larger than 300 nm. Second, the cage must exhibit a high uncaging quantum yield together with a clean photoreaction without unwanted byproducts. Third, the PPG must be soluble in the medium for the application target, i.e., water and buffer solutions for biochemical experiments. Fourth, it is important that the uncaging reaction does not produce any toxic product, and the absorption behavior of the photoproduct should not be identical to that of the cage itself to avoid competitive absorption. All together, fulfilling these requirements poses a difficult task for both experimentalists and theoreticians. The quest for efficient new photocages thus requires detailed understanding of the underlying reaction mechanism to devise reasonable structure-activity relationships. Once this mechanism is clearly understood, it would in principle be possible to tune the decisive molecular properties of candidate molecules. To effectively carry out this task, new photocages can be studied by computational means, providing helpful guidelines as to which scaffold and substituent combinations are worth synthesizing. The

---

Parts of this chapter have already been published in:

- V. Hermanns, M. Scheurer, N. Kersten, C. Abdellaoui, J. Wachtveitl, A. Dreuw, and A. Heckel, "Re-Thinking Uncaging: A New Antiaromatic Photocage Driven by a Gain of Resonance Energy", *Chem. Eur. J.* **2021**, Accepted Author Manuscript. DOI: 10.1002/chem.202102351. (Reference 273) design of new photocages relies on common molecular scaffolds,[274] and a species that has only recently gained attention for this specific use is fluoren-9-ol.[276,277] The photochemical solvolysis of fluoren-9-ol (FOH) was first reported in 1985,[278] and follow-up studies revealed that the intermediate species of this photoreaction is quenched through nucleophilic solvent molecules.[279] Correspondingly, the diagram of the reaction steps is shown in Figure 9.1. Through photoinduced heterolysis of fluoren-9-ol, a hydroxide anion OH^- is detached, forming a fluorenyl cation intermediate species F^+.[276,280] In the final step, the cationic intermediate is deactivated through nucleophilic attack of a methanol (MeOH) solvent molecule. The first practical application of photocages based on this mechanism was shown in 2018,[277] i.e., the first-generation of fluorene-based PPGs. Especially biologically relevant leaving groups could be efficiently released from these cages. Despite this success, there is still room for systematic improvement and a more detailed understanding of fluorene-based photocages. Extending the toolbox of cages based on fluorene requires an efficient computational screening procedure to discern potent candidate molecules from inefficient ones. Through a well-defined structure-activity relationship, not established yet for this type of photocage, it would be possible to design new molecular scaffolds with even larger UV/Vis absorption red shifts, clean photoreactions, and synthetic variability. An *in silico* procedure relying on, e.g., black-box DFT calculations would provide a quantitative perspective on how to improve fluorene-based photocages systematically. The most promising candidate molecules arising from such an undertaking can then be synthesized and tested *in vitro*. Inspired by the work from Winter and co-workers,[281] I devised a new screening procedure as proposed above, ultimately leading to an improved, second-generation photocage based on fluorenol.[273] This was possible through a joint experimental and computational endeavor with V. Hermanns from the Heckel group at the University of Frankfurt. In the next section, I first analyze the proposed uncaging mechanism of fluoren-9-ol in detail. With the confirmed mechanism as a basis, I explain the simplistic screening procedure which can be applied to rapidly pre-scan and assess the quality of new fluorene-like photocages at a minimal computational cost.

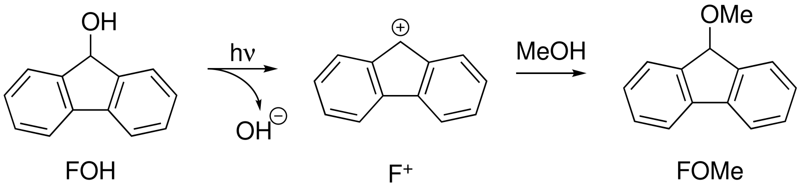


Figure 9.1: Photolysis reaction of fluoren-9-ol and quenching through the solvent MeOH.

9.1 Uncaging Mechanism of Fluorenol

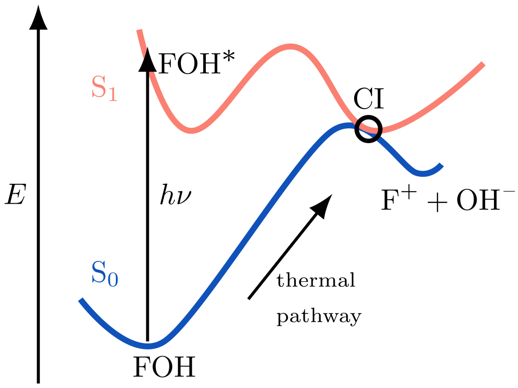


Figure 9.2: Hypothetical reaction mechanism for photoheterolysis of fluorene-like compounds.[281]

According to Winter and co-workers, the photoheterolysis of fluorene-like compounds and other species is governed by a conical intersection (CI) between the ground state and the energetically lowest singlet excited state.[281] This proposed mechanism is illustrated in Figure 9.2. After photoexcitation from the singlet ground state S_0 to the first singlet excited state S_1, the reaction path proceeds via a transition state, however, not directly to the minimum product state, but through a CI energetically close to the S_0 minimum of the heterolysis product $F^+ + OH^-$. The thermal pathway is, in the depicted scenario, rather unfavorable and unlikely to occur. To corroborate the proposed and widely accepted reaction mechanism by a theoretical analysis, I carried out DFT and TDDFT calculations at the CAM-B3LYP/def2-SVP level of theory for the simplest fluoren-9-ol species.[127,282] These computations were carried out using version 5.3 of the Q-Chem program package.[93,94] First, an optimization of the S_1 excited state was run with TD-CAM-B3LYP/def2-SVP in the TDA approximation[144] to obtain the educt structure of the photoreaction. In the following step, the structure of the CI was sought by means of a minimum energy crossing point (MECP) optimization between the S_0 and S_1 state, using the spin-flip (SF) formalism as implemented in Q-Chem.[283] The obtained molecular structure of the CI indeed showed that the OH^- moiety was cleaved from the fluorene part. As a final step, the transition state (TS) of this reaction path was optimized, in which the OH^- was partly detached from fluorene. The results of this reaction mechanism study are illustrated in Figure 9.3, together with the relative energies of the involved states. The energy difference between the Franck-Condon point and the S_1 minimum is 0.48 eV, such that enough excess energy should in principle be available to overcome the corresponding

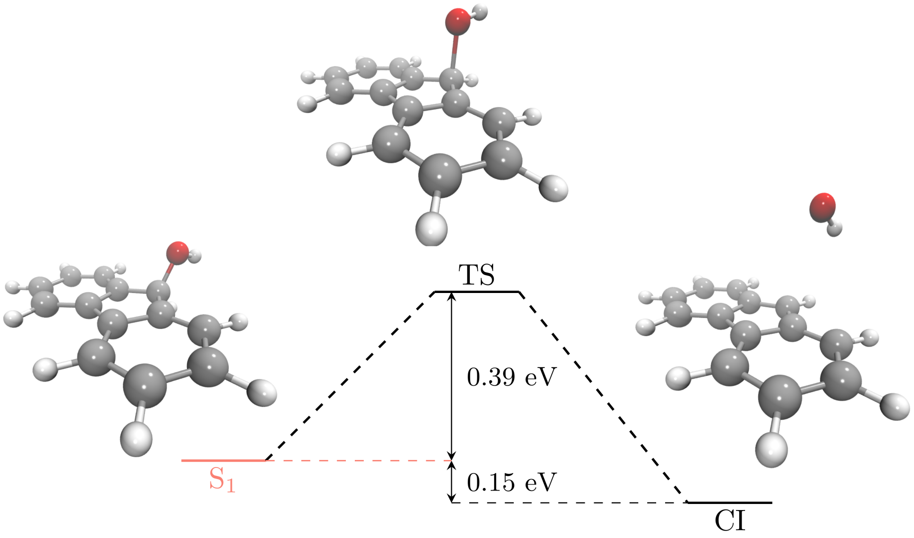


Figure 9.3: Illustration of species involved in the photoheterolysis reaction, including relative energies.

barrier of 0.39 eV. The C7-O distance steadily increases from 1.44 Å in the S_1 state, to 1.85 Å in the TS, to 3.15 Å in the CI structure. As follows, the C7-O bond is already broken at the CI, indicating proximity to the product state $F^+ + OH^-$ in the target S_0 state. This analysis confirms for the first time that the photoheterolysis of fluorene-like species can in principle proceed via a productive CI channel, which is crucial for understanding and designing improved fluorene-derived photocages. If existence of such a CI channel is mandatory, it should be a reliable indicator whether a certain compound exhibits a productive uncaging behavior or not. As a matter of fact, two scenarios are possible for such a reaction type, illustrated in Figure 9.4. The key part considered here is the relative energy of the S_0 and S_1 state at the product geometry.[281] For a high-energy $F^+ + OH^-$ species, the two potential energy surfaces are energetically close to each other (see Fig. 9.4a), i.e., the ground state cation is destabilized. Note that the effect of the leaving group (LG), in this case OH^-, is completely neglected in this line of reasoning. The proximity of the S_0/S_1 PES, however, makes existence of a nearby CI rather likely. On the contrary, the involved PES are further apart when a low-energy product state is present (see Fig. 9.4b): In that case, the cationic species is stabilized, and the energy gap between the PES becomes quite large. As a result, the productive uncaging CI channel is likely not existent in the stabilized cation case. From a practical point of view, it is rather time-consuming to use the existence of a productive CI directly as a parameter for efficient uncaging, because it is not as easily implementable in an automated manner. The vertical excitation energy of the cationic species ΔE_{vert} can, however, be used as an

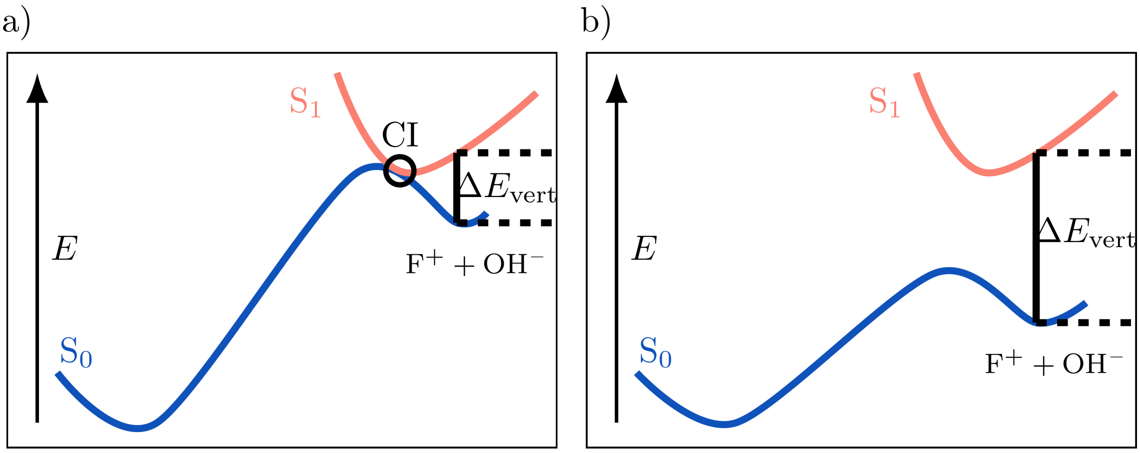


Figure 9.4: Schematic illustration of ground and excited state potential energy surfaces for a) destabilized fluorenyl cations and b) stabilized fluorenyl cations.[281]

indirect indicator for a nearby CI. This is visible directly from the illustration in Figure 9.4: The lower the vertical excitation energy, the closer are the involved potential energy surfaces, and consequently, existence of a productive CI is likely. In the other scenario, the larger the vertical excitation energy, the larger the energy gap between the S_0 and S_1 PES, and the productive channel is most likely not existent, and the product state is photochemically inaccessible. As such, a cationic species with a small vertical excitation energy should, in principle, be suited as efficient fluorene-based photocage. Of note, the absolute vertical excitation energies are not helpful. Much rather, a ranking of compounds can be created to opt for the ideal molecular species in this regard. Using the vertical excitation energy as an indicator for efficient uncaging greatly simplifies the computational screening procedure. It essentially boils down to a two-step task, in which the cationic species is first optimized in its electronic ground state, and the excitation energies are then computed for the optimized structure. This task can be fully automated, and solvent effects can be conveniently included through a continuum model. A major shortcoming of the computational assessment is that the effect of the leaving group is completely ignored, which could in principle have a large effect on the uncaging efficiency. Note that this simplification makes it possible in the first place to study the plain scaffolds of prospect fluorene-based cages without taking into account the vast amount of possible cage/leaving group combinations. One could say that the protocol established here judges the quality of the cage itself, independent of the attached leaving group.

9.2 Computational Screening and Ranking of New Fluorene Photocages

To improve upon the first-generation fluorenol photocages, modifications to the existing cages in order to destabilize the cation were discussed, using chemical intuition. Not only can the substituents at the ring systems be varied[277] but even more changes in the electronic structure can be introduced through heteroatoms in the ring system. For this reason, we proposed so-called “aza-fluorenol” derivatives, where a nitrogen atom is placed in each of the outer aromatic rings of the fluorene derivative. These molecules were, however, synthetically not accessible and did not provide much flexibility for interchanging ring substituents. Considering the nitrogen fluorenol derivatives a dead end, we came up with the idea to replace the six-membered benzene rings with thiophene moieties. These cyclopenta-dithiophene derivatives are of course not derivatives of fluorene itself, but through the similar structure can be considered fluorene-like. Three different positions of the sulfur atoms were suggested, leading to ortho-, meta-, and para-sulfur-cyclopenta-dithiophene scaffolds. The position labels ortho, meta, and para refer to the position of the sulfur atom from the top ring position, connected to the *ipso*-C atom.[273] Using standard nomenclature, these would be denoted $7H$, $6H$, and $4H$ derivatives, respectively. The proposed scaffolds in their simplest form were then tested with the screening protocol outlined above. All computations were performed using Q-Chem 5.3.[93,94] Cationic species were optimized at the CAM-B3LYP/def2-SVP level of theory[127,282] employing a polarizable continuum model for water ($\varepsilon = 78.4$, $n^2 = 1.76$).[284,285] Afterwards, the five energetically

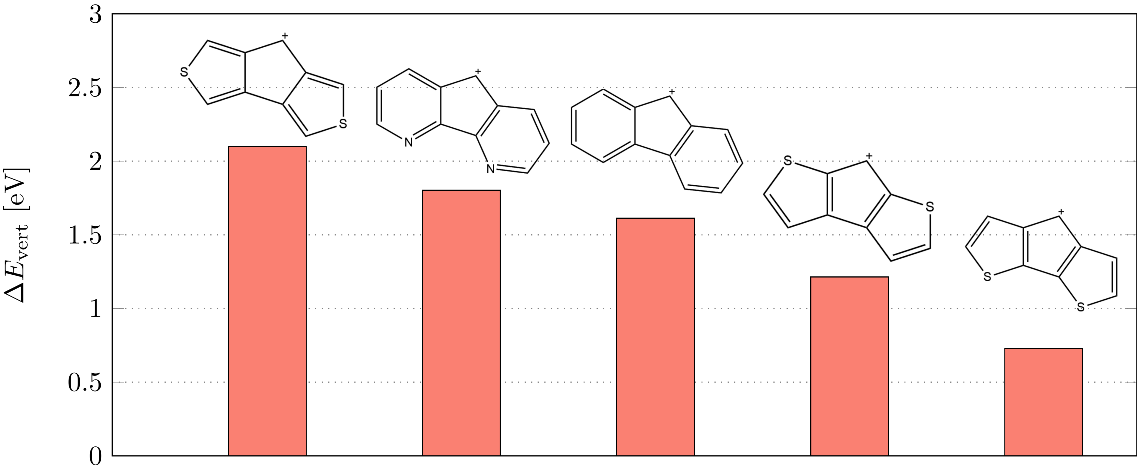


Figure 9.5: Ranking of selected fluorene scaffolds by the $S_0 \rightarrow S_1$ vertical excitation energies ΔE_{vert} of the cationic species. Data reproduced from Ref. 273.

lowest singlet excited states were computed with time-dependent DFT at the same level of theory, from which the lowest vertical excitation energy ΔE_{vert} was extracted. The corresponding ranking by decreasing ΔE_{vert} value for the raw fluorene-like scaffolds is shown in Figure 9.5.[273] One can see that the cationic intermediate of the bare fluorenol scaffold has a relatively high vertical excitation energy of about 1.61 eV. Compared to this, the para-sulfur-cyclopenta-dithiophene derivative showed a much smaller excitation energy of only 0.73 eV. The corresponding meta and ortho derivatives also showed higher excitation energies compared to the para species. The aza-derivative was added here for completeness, showing a slightly larger excitation energy than the plain fluorenol scaffold. These five molecules are of course just the unmodified ring systems, and different substitution patterns have not been taken into account. Nevertheless, the para-sulfur-cyclopenta-dithiophene derivative seems to be the most promising candidate when following the newly established screening and ranking protocol. With these assessments at hand, my collaborator synthesized different variations of the para-sulfur-cyclopenta-dithiophene, also varying the ring substituents.[273] The resulting compounds were then analyzed with respect to UV/Vis absorption/fluorescence behavior and the uncaging quantum yield was determined. Intriguingly, the para-sulfur-cyclopenta-dithiophene without any further substituents reached an enormously high uncaging quantum yield of 28%.[273] The compound with the most favorable, red-shifted UV/Vis absorption bands still showed a remarkable uncaging quantum yield of approximately 5%, compared to similar photocages with a maximum absorption around 400 nm.[273,275] With this new generation of sulfur-based fluorene-like PPGs, the effective release of the biologically relevant neurotransmitter serotonin under physiological buffer conditions could be demonstrated, too.[273]

9.3 Conclusions

In this chapter, I presented the first efficient computational protocol to design and improve new photocages based on fluorene. The prospect quality of a new compound can be probed *in silico* directly through the vertical absorption energy of the energetically lowest singlet excited state of the intermediate cationic species. Furthermore, the proposed reaction pathway was confirmed for the first time by computational means. Understanding the mechanism was also the cornerstone for deriving the simplified protocol, i.e., the vertical absorption energy serves as an indicator for a nearby conical intersection through which the photoheterolysis pathway proceeds, thereby detaching the leaving group. The computational protocol is not only conclusive from a theoretical perspective, but was also successfully applied in practice to design the next-generation fluorene-based photocages recently.[273] In this work, cyclopenta-dithiophene derivatives, the computationally determined ideal compound candidates, showed improved UV/Vis absorption behavior and uncaging quantum yields, compared to previously proposed fluorene-based PPGs.[277] Even though the computational protocol may seem simple, it has proven practical reliability and can thus be considered indispensable for further improving this promising family of photocages in the future.

Chapter 10

Conclusions

In this thesis, I presented new quantum chemical methodologies to accurately model molecular properties in complex environments. I especially focused on the development of hybrid quantum-classical approaches for computational spectroscopy simulations. For this purpose, I utilized the algebraic-diagrammatic construction (ADC) scheme for the polarization propagator and combined it with the polarizable embedding (PE) model, which is capable of describing mutual polarization effects between the quantum region and environment in a fully self-consistent manner.

My first PE-ADC combination employed a standard ADC scheme on top of a self-consistent PE-HF ground state, thus only including explicit environment effects through the "solvated" molecular orbitals. To improve the excitation energies produced by this pt-PE-ADC scheme, I exploited perturbative corrections, accounting for the interaction of the excited state wave function with the environment and the non-resonant excitonic coupling. The pt-PE-ADC scheme is computationally inexpensive since the only additional task is the evaluation of *a posteriori* corrections. A major shortcoming, however, is that no dynamic response of the polarizable environment during the correlation treatment is included. The second combination, LR-PE-ADC, includes a dynamic coupling term in the ADC secular matrix evaluated from the zeroth-order transition density of the respective ADC scheme. This iterative treatment of environment coupling was suitable for modeling *any* kind of molecular response property within the ADC/ISR framework. For exploration of potential energy surfaces with polarizable environments, I derived all working equations for PE-ADC analytic nuclear gradients. In future work, PE could be combined with related Hermitian electronic structure methods, for instance unitary coupled cluster (UCC).[286] Moreover, the theoretical derivations for LR-PE-ADC could be transferred to other implicit or explicit polarizable models, e.g., the PCM or EFP method. This is possible because only the underlying expressions for the electrostatic and polarization interactions have to be modified, but the general structure among those methods is almost identical.[23]

For efficient implementation of the derived approaches, I initially developed the standalone CPPE library for the PE model. Relying on hybrid Python/C++ code, I interfaced the CPPE library to four quantum chemical program packages, three of them can be freely used. This made the PE model available in more programs than ever before. The CPPE library can be easily built from sources or installed from pre-compiled binaries deposited on popular package repositories. The CPPE library will enable the scientific community to create combinations with many quantum chemical methods. As a next step, I added a general interface for environment models to the `adcc` toolkit. Using this infrastructure, I implemented and tested the pt-PE-ADC and LR-PE-ADC approaches. Furthermore, I made correlated ground and excited state gradients for MP and ADC up to second order available in `adcc`, which can be used together with PE. In future work, the performance of this gradient implementation could be further improved in order to allow for treatment of larger systems. In addition, continuum solvation models would complement the rich feature set of environment models I added to `adcc`, providing even more flexibility for users. I developed the `respondo` library as a plugin for `adcc`, which contains the necessary building blocks to evaluate a variety of response functions in the ADC/ISR framework, each combinable with PE, and made ADC/ISR response properties freely available for the first time. Most importantly, my combined PE-ADC schemes are available for all ADC variants in `adcc`, and all response properties in `respondo`.

In the presented test case on water-solvated pNA, the errors for excitation energies of the pt-PE-ADC scheme were negligible, and bulk solvent effects were accurately captured. Using pt-PE-ADC showed that the polarizable protein environment efficiently promotes a CT excitation pivotal for the photoprotection mechanism of dodecin. This case study proved that the pt-PE-ADC scheme can be applied to model excitation processes in large biomolecular environments. Next, I benchmarked the LR-PE-ADC approach for excitation energies, oscillator strengths, and TPA strengths for small, water-solvated chromophores. I showed that LR-PE-ADC combined with two state-of-the-art add-ons to fix electron spill-out artifacts and intensities is clearly superior to commonly employed hybrid schemes to compute higher-order properties in polarizable environments. Finally, I tested the new PE-MP and PE-ADC gradient implementations by comparing to finite differences. Future applications of PE-ADC schemes could include modeling X-ray processes,[287] or ionization/electron attachment in the condensed phase.[288,289] In addition, more benchmark studies should be carried out for higher-order response properties on a heterogeneous set of test systems. Ultimately, it might be possible to investigate more light-driven biomolecular systems, similar to dodecin, to entirely reap the benefits of the combined methods I presented.

I showed a way to generally improve the performance of PE models: The iterative solution of the classical induced dipole equations normally exhibits an asymptotic quadratic-scaling behavior with the number of polarizable sites. Using the fast multipole method (FMM), the scaling becomes linear and the performance bottleneck of the classical part of PE schemes is removed. I achieved the PE-FMM implementation through auto-generated code from a third-party library and included it in CPPE, automatically available in all connected host programs. I proved that the errors introduced through FMM-based electric field computations are well controllable through the model parameters and tested the error propagation for pt-PE-ADC excitation energies. Remarkably, I found that the error introduced through FMM is negligible for such properties, but enormous speed-ups compared to conventional PE formulations were achieved already for a few thousand polarizable sites. A flagship test run on a system with more than one million polarizable sites illustrated that my PE-FMM scheme is capable of treating sizable polarizable environments efficiently.

I presented two algorithms for solving general response equations in the ADC/ISR framework, which rely on a subspace projection approach together with an efficient block folding scheme in the case of ADC(2). I analyzed the convergence of the new solvers and compared it to standard numerical approaches. The newly implemented algorithms converged in numerically difficult cases without any problems. Naturally, these numerically stable schemes can be used for PE-ADC response equations, too. Furthermore, I showed comprehensive numerical case studies for complex excited state polarizabilities in the ADC/ISR framework and studied their performance in comparison to related wave function methods. It would be interesting to compare ADC response properties based on energy derivatives with the ISR-based approach in the future.

I developed two new approaches to model properties under extreme environment conditions. The first approach is the GOSTSHYP method which applies hydrostatic pressure to a molecular system via Gaussian potentials on the solvent-accessible surface. GOSTSHYP is the first pressure model with the capability to run geometry optimizations and dynamics simulations at a pre-defined pressure. It works for atoms and molecular systems, and mediates hydrostatic pressure through direct compression of the electron density. I applied GOSTSHYP to describe a pressure-induced Diels-Alder reaction. For improved performance, I proposed a simplification of GOSTSHYP, together with grid-based evaluation of the pressure potential. Future developments on GOSTSHYP should include direct contributions through the surrounding solvent. A combination of GOSTSHYP with an excited state method would enable the investigation of spectroscopic properties under pressure in future research. The second approach is a computational protocol based on steered molecular dynamics (SMD) simulations and the quantum chemical Judgement of Energy DIstribution (JEDI) analysis. Using this protocol, I showed that the mechanical lability of rubredoxin is not caused by hydrogen bond networks, as previously assumed in the literature. While it was not possible to determine which electronic and structural parameters of the system explain the low rupture force, the SMD-JEDI protocol will enable similar investigations of force-induced processes in biomolecular systems. It might be feasible to devise a more compact and direct hybrid quantum-classical workflow including the JEDI analysis. This would require well-converged structures and reliable analytical nuclear Hessian matrices.

I established an efficient and predictive screening procedure for fluorene-like molecular scaffolds to judge their suitability as photocages. Together with an experimental collaborator, we found derivatives of cyclopenta-dithiophene to outperform existing fluorene-based photocages with respect to absorption behavior and uncaging quantum yield, resulting in the next-generation photocages of this type. Hence, the computational screening procedure proved helpful in practice and will support the quest for even more efficient fluorene-like photocages in the future.

All in all, I hope that especially the PE-ADC methodologies developed in this thesis will provide a helpful starting point for future development and applications. Through the availability in open-source toolkits, I expect that the PE model will be more routinely used and that a sustainable environment for modeling molecular properties in complex environments will be created eventually.

Bibliography

[1] Barone, V. *Computational Strategies for Spectroscopy*; Wiley, 2011.

[2] Norman, P.; Ruud, K.; Saue, T. *Principles and Practices of Molecular Properties*; John Wiley & Sons, Ltd, 2018.

[3] González, L.; Lindh, R. *Quantum Chemistry and Dynamics of Excited States*; Wiley, 2020.

[4] Warshel, A.; Levitt, M. Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *J. Mol. Biol.* **1976**, *103*, 227–249.

[5] Senn, H. M.; Thiel, W. QM/MM Methods for Biomolecular Systems. *Angew. Chem. Int. Ed.* **2009**, *48*, 1198–1229.

[6] Melo, M. C. R.; Bernardi, R. C.; Rudack, T.; Scheurer, M.; Riplinger, C.; Phillips, J. C.; Maia, J. D. C.; Rocha, G. B.; Ribeiro, J. V.; Stone, J. E.; Neese, F.; Schulten, K.; Luthey-Schulten, Z. NAMD goes quantum: An integrative suite for hybrid simulations. *Nat Methods* **2018**, *15*, 351–354.

[7] Mennucci, B.; Corni, S. Multiscale modelling of photoinduced processes in composite systems. *Nat Rev Chem* **2019**, *3*, 315–330.

[8] Nottoli, M.; Cupellini, L.; Lipparini, F.; Granucci, G.; Mennucci, B. Multiscale Models for Light-Driven Processes. *Annu. Rev. Phys. Chem.* **2021**, *72*, 489–513.

[9] Andruniów, T.; Olivucci, M. *QM/MM Studies of Light-responsive Biological Systems*; Springer International Publishing, 2021.

[10] Klamt, A.; Schüürmann, G. COSMO: A new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *J. Chem. Soc., Perkin Trans. 2* **1993**, 799–805.

[11] Tomasi, J.; Mennucci, B.; Cammi, R. Quantum Mechanical Continuum Solvation Models. *Chem. Rev.* **2005**, *105*, 2999–3094.

[12] Mennucci, B. Polarizable continuum model. *WIREs Comput Mol Sci* **2012**, *2*, 386–404.

[13] Mennucci, B.; Tomasi, J.; Cammi, R.; Cheeseman, J.; Frisch, M.; Devlin, F.; Gabriel, S.; Stephens, P. Polarizable Continuum Model (PCM) Calculations of Solvent Effects on Optical Rotations of Chiral Molecules. *J. Phys. Chem. A* **2002**, *106*, 6102–6113.

[14] Olsen, J. M.; Aidas, K.; Kongsted, J. Excited States in Solution through Polarizable Embedding. *J. Chem. Theory Comput.* **2010**, *6*, 3721–3734.

[15] Olsen, J. M. H.; Kongsted, J. *Adv. Quantum Chem.*; 2011; Vol. 61; pp 107–143.

[16] List, N. H.; Olsen, J. M. H.; Kongsted, J. Excited states in large molecular systems through polarizable embedding. *Phys. Chem. Chem. Phys.* **2016**, *18*, 20234–20250.

[17] Loco, D.; Polack, E.; Caprasecca, S.; Lagardère, L.; Lipparini, F.; Piquemal, J.-P.; Mennucci, B. A QM/MM Approach Using the AMOEBA Polarizable Embedding: From Ground State Energies to Electronic Excitations. *J. Chem. Theory Comput.* **2016**, *12*, 3654–3661.

[18] Steinmann, C.; Reinholdt, P.; Nørby, M. S.; Kongsted, J.; Olsen, J. M. H. Response properties of embedded molecules through the polarizable embedding model. *Int J Quantum Chem* **2018**, *119*, e25717.

[19] Loco, D.; Lagardère, L.; Cisneros, G. A.; Scalmani, G.; Frisch, M.; Lipparini, F.; Mennucci, B.; Piquemal, J.-P. Towards large scale hybrid QM/MM dynamics of complex systems with advanced point dipole polarizable embeddings. *Chem. Sci.* **2019**, *10*, 7200–7211.

[20] Bondanza, M.; Nottoli, M.; Cupellini, L.; Lipparini, F.; Mennucci, B. Polarizable embedding QM/MM: The future gold standard for complex (bio)systems? *Phys. Chem. Chem. Phys.* **2020**, *22*, 14433–14448.

[21] Loco, D.; Lagardère, L.; Adjoua, O.; Piquemal, J.-P. Atomistic Polarizable Embeddings: Energy, Dynamics, Spectroscopy, and Reactivity. *Acc. Chem. Res.* **2021**, *54*, 2812–2822.

[22] Hartmann, P.; Reinholdt, P.; Kongsted, J. In *Challenges and Advances in Computational Chemistry and Physics*; Andruniów, T., Olivucci, M., Eds.; Springer International Publishing: Cham, 2020; pp 143–195.

[23] Nottoli, M.; Lipparini, F. General formulation of polarizable embedding models and of their coupling. *J. Chem. Phys.* **2020**, *153*, 224108.

[24] Dreuw, A.; Head-Gordon, M. Failure of Time-Dependent Density Functional Theory for Long-Range Charge-Transfer Excited States: The Zincbacteriochlorin−Bacteriochlorin and Bacteriochlorophyll−Spheroidene Complexes. *J. Am. Chem. Soc.* **2004**, *126*, 4007–4016.

[25] Dreuw, A.; Head-Gordon, M. Single-Reference ab Initio Methods for the Calculation of Excited States of Large Molecules. *Chem. Rev.* **2005**, *105*, 4009–4037.

[26] Schirmer, J.; Dreuw, A. Critique of the foundations of time-dependent density-functional theory. *Phys. Rev. A* **2007**, *75*, 022513.

[27] Sneskov, K.; Schwabe, T.; Kongsted, J.; Christiansen, O. The polarizable embedding coupled cluster method. *J. Chem. Phys.* **2011**, *134*, 104108.

[28] Schwabe, T.; Sneskov, K.; Haugaard Olsen, J. M.; Kongsted, J.; Christiansen, O.; Hättig, C. PERI–CC2: A Polarizable Embedded RI-CC2 Method. *J. Chem. Theory Comput.* **2012**, *8*, 3274–3283.

[29] Hršak, D.; Marefat Khah, A.; Christiansen, O.; Hättig, C. Polarizable Embedded RI-CC2 Method for Two-Photon Absorption Calculations. *J. Chem. Theory Comput.* **2015**, *11*, 3669–3678.

[30] Eriksen, J. J.; Sauer, S. P.; Mikkelsen, K. V.; Jensen, H. J.; Kongsted, J. On the importance of excited state dynamic response electron correlation in polarizable embedding methods. *J. Comput. Chem.* **2012**, *33*, 2012–2022.

[31] Schirmer, J. Beyond the random-phase approximation: A new approximation scheme for the polarization propagator. *Phys. Rev. A* **1982**, *26*, 2395–2416.

[32] Marefat Khah, A.; Karbalaei Khani, S.; Hättig, C. Analytic Excited State Gradients for the QM/MM Polarizable Embedded Second-Order Algebraic Diagrammatic Construction for the Polarization Propagator PE-ADC(2). *J. Chem. Theory Comput.* **2018**, *14*, 4640–4650.

[33] Sen, R.; Dreuw, A.; Faraji, S. Algebraic diagrammatic construction for the polarisation propagator in combination with effective fragment potentials. *Phys. Chem. Chem. Phys.* **2019**, *21*, 3683–3694.

[34] Wormit, M.; Rehn, D. R.; Harbach, P. H.; Wenzel, J.; Krauter, C. M.; Epifanovsky, E.; Dreuw, A. Investigating excited electronic states using the algebraic diagrammatic construction (ADC) approach of the polarisation propagator. *Mol. Phys.* **2014**, *112*, 774–784.

[35] Dreuw, A.; Wormit, M. The algebraic diagrammatic construction scheme for the polarization propagator for the calculation of excited states. *WIREs Comput Mol Sci* **2014**, *5*, 82–95.

[36] Dreuw, A. The Algebraic-Diagrammatic Construction Scheme for the Polarization Propagator. *Quantum Chemistry and Dynamics of Excited States: Methods and Applications* **2020**, 109–131.

[37] Knippenberg, S.; Rehn, D.; Wormit, M.; Starcke, J.; Rusakova, I.; Trofimov, A.; Dreuw, A. Calculations of nonlinear response properties using the intermediate state representation and the algebraic-diagrammatic construction polarization propagator approach: Two-photon absorption spectra. *J. Chem. Phys.* **2012**, *136*, 064107.

[38] Smith, D. G.; Burns, L. A.; Simmonett, A. C.; Parrish, R. M.; Schieber, M. C.; Galvelis, R.; Kraus, P.; Kruse, H.; Di Remigio, R.; Alenaizan, A.; James, A. M.; Lehtola, S.; Misiewicz, J. P.; Scheurer, M.; Shaw, R. A.; Schriber, J. B.; Xie, Y.; Glick, Z. L.; Sirianni, D. A.; O'Brien, J. S.; Waldrop, J. M.; Kumar, A.; Hohenstein, E. G.; Pritchard, B. P.; Brooks, B. R.; Schaefer, H. F.; Sokolov, A. Y.;

Patkowski, K.; DePrince, A. E.; Bozkaya, U.; King, R. A.; Evangelista, F. A.; Turney, J. M.; Crawford, T. D.; Sherrill, C. D. Psi4 1.4: Open-source software for high-throughput quantum chemistry. *J. Chem. Phys.* **2020**, *152*, 184108.

[39] Sun, Q.; Zhang, X.; Banerjee, S.; Bao, P.; Barbry, M.; Blunt, N. S.; Bogdanov, N. A.; Booth, G. H.; Chen, J.; Cui, Z.-H.; Eriksen, J. J.; Gao, Y.; Guo, S.; Hermann, J.; Hermes, M. R.; Koh, K.; Koval, P.; Lehtola, S.; Li, Z.; Liu, J.; Mardirossian, N.; McClain, J. D.; Motta, M.; Mussard, B.; Pham, H. Q.; Pulkin, A.; Purwanto, W.; Robinson, P. J.; Ronca, E.; Sayfutyarova, E. R.; Scheurer, M.; Schurkus, H. F.; Smith, J. E.; Sun, C.; Sun, S.-N.; Upadhyay, S.; Wagner, L. K.; Wang, X.; White, A.; Whitfield, J. D.; Williamson, M. J.; Wouters, S.; Yang, J.; Yu, J. M.; Zhu, T.; Berkelbach, T. C.; Sharma, S.; Sokolov, A. Y.; Chan, G. K.-L. Recent developments in the PySCF program package. *J. Chem. Phys.* **2020**, *153*, 024109.

[40] Schirmer, J.; Trofimov, A. Intermediate state representation approach to physical properties of electronically excited molecules. *J. Chem. Phys.* **2004**, *120*, 11449–11464.

[41] Scheurer, M.; Fransson, T.; Norman, P.; Dreuw, A.; Rehn, D. R. Complex excited state polarizabilities in the ADC/ISR framework. *J. Chem. Phys.* **2020**, *153*, 074112.

[42] Scheurer, M. *Polarizable Embedding for the Algebraic-Diagrammatic Construction Scheme*; Springer Fachmedien Wiesbaden, 2020.

[43] Shavitt, I.; Bartlett, R. J. *Many – Body Methods in Chemistry and Physics*; Cambridge University Press, 2009.

[44] Helgaker, T.; Jørgensen, P.; Olsen, J. *Molecular Electronic-Structure Theory*; John Wiley & Sons, Ltd, 2000.

[45] Hartree, D. The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods. *Math. Proc. Camb. Phil. Soc.* **1928**, *24*, 89–110.

[46] Hartree, D.; Hartree, W. Self-consistent field, with exchange, for beryllium. *Proc. R. Soc. Lond. A* **1935**, *150*, 9–33.

[47] Jensen, F. *Introduction to computational chemistry*; John wiley & sons, 2017.

[48] Harbach, P. H.; Wormit, M.; Dreuw, A. The third-order algebraic diagrammatic construction method (ADC(3)) for the polarization propagator for closed-shell molecules: Efficient implementation and benchmarking. *J. Chem. Phys.* **2014**, *141*, 064113.

[49] Dempwolff, A. L. Development and Implementation of High-Level Propagator Methods for the Description of Electronically Stable and Unstable States. Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg, 2020.

[50] Helgaker, T.; Coriani, S.; Jørgensen, P.; Kristensen, K.; Olsen, J.; Ruud, K. Recent Advances in Wave Function-Based Methods of Molecular-Property Calculations. *Chem. Rev.* **2012**, *112*, 543–631.

[51] Trofimov, A.; Krivdina, I.; Weller, J.; Schirmer, J. Algebraic-diagrammatic construction propagator approach to molecular response properties. *Chem. Phys.* **2006**, *329*, 1–10.

[52] Fransson, T.; Rehn, D. R.; Dreuw, A.; Norman, P. Static polarizabilities and C6 dispersion coefficients using the algebraic-diagrammatic construction scheme for the complex polarization propagator. *J. Chem. Phys.* **2017**, *146*, 094301.

[53] Hodecker, M.; Rehn, D. R.; Norman, P.; Dreuw, A. Algebraic-diagrammatic construction scheme for the polarization propagator including ground-state coupled-cluster amplitudes. II. Static polarizabilities. *J. Chem. Phys.* **2019**, *150*, 174105.

[54] Rehn, D. R.; Dreuw, A.; Norman, P. Resonant Inelastic X-ray Scattering Amplitudes and Cross Sections in the Algebraic Diagrammatic Construction/Intermediate State Representation (ADC/ISR) Approach. *J. Chem. Theory Comput.* **2017**, *13*, 5552–5559.

[55] Fetter, A. L.; Walecka, J. D. *Quantum theory of many-particle systems*; Courier Corporation, 2012.

[56] Kauczor, J.; Norman, P.; Christiansen, O.; Coriani, S. Communication: A reduced-space algorithm for the solution of the complex linear response equations used in coupled cluster damped response theory. *J. Chem. Phys.* **2013**, *139*, 211102.

[57] Stone, A. *The Theory of Intermolecular Forces*; Oxford University Press, 2013.

[58] Schwabe, T. General theory for environmental effects on (vertical) electronic excitation energies. *J. Chem. Phys.* **2016**, *145*, 154105.

[59] List, N. H.; Beerepoot, M. T.; Olsen, J. M. H.; Gao, B.; Ruud, K.; Jensen, H. J. A.; Kongsted, J. Molecular quantum mechanical gradients within the polarizable embedding approach—Application to the internal vibrational Stark shift of acetophenone. *J. Chem. Phys.* **2015**, *142*, 034119.

[60] Applequist, J.; Carl, J. R.; Fung, K.-K. Atom dipole interaction model for molecular polarizability. Application to polyatomic molecules and determination of atom polarizabilities. *J. Am. Chem. Soc.* **1972**, *94*, 2952–2960.

[61] Scheurer, M.; Herbst, M. F.; Reinholdt, P.; Olsen, J. M. H.; Dreuw, A.; Kongsted, J. Polarizable Embedding Combined with the Algebraic Diagrammatic Construction: Tackling Excited States in Biomolecular Systems. *J. Chem. Theory Comput.* **2018**, *14*, 4870–4883.

[62] del Valle, F.; Tomasi, J. Electron correlation and solvation effects. I. Basic formulation and preliminary attempt to include the electron correlation in the quantum mechanical polarizable continuum model so as to study solvation phenomena. *Chem. Phys.* **1991**, *150*, 139–150.

[63] Ángyán, J. G. Choosing between alternative MP2 algorithms in the self-consistent reaction field theory of solvent effects. *Chem. Phys. Lett.* **1995**, *241*, 51–56.

[64] Slipchenko, L. V. Solvation of the Excited States of Chromophores in Polarizable Environment: Orbital Relaxation versus Polarization†. *J. Phys. Chem. A* **2010**, *114*, 8824–8830.

[65] Lunkenheimer, B.; Köhn, A. Solvent Effects on Electronically Excited States Using the Conductor-Like Screening Model and the Second-Order Correlated Method ADC(2). *J. Chem. Theory Comput.* **2012**, *9*, 977–994.

[66] Schröder, H.; Schwabe, T. Corrected Polarizable Embedding: Improving the Induction Contribution to Perichromism for Linear Response Theory. *J. Chem. Theory Comput.* **2018**, *14*, 833–842.

[67] Karbalaei Khani, S.; Marefat Khah, A.; Hättig, C. COSMO-RI-ADC(2) excitation energies and excited state gradients. *Phys. Chem. Chem. Phys.* **2018**, *20*, 16354–16363.

[68] Caricato, M.; Mennucci, B.; Tomasi, J.; Ingrosso, F.; Cammi, R.; Corni, S.; Scalmani, G. Formation and relaxation of excited states in solution: A new time dependent polarizable continuum model based on time dependent density functional theory. *J. Chem. Phys.* **2006**, *124*, 124520.

[69] Hodecker, M.; Rehn, D. R.; Dreuw, A.; Höfener, S. Similarities and differences of the Lagrange formalism and the intermediate state representation in the treatment of molecular properties. *J. Chem. Phys.* **2019**, *150*, 164125.

[70] Nanda, K. D.; Krylov, A. I. The effect of polarizable environment on two-photon absorption cross sections characterized by the equation-of-motion coupled-cluster singles and doubles method combined with the effective fragment potential approach. *J. Chem. Phys.* **2018**, *149*, 164109.

[71] Fu, M.; Wesolowski, T. A. The Challenge of Accurate Computation of Two-Photon Absorption Properties of Organic Chromophores in the Condensed Phase. *J. Chem. Theory Comput.* **2021**, *17*, 3652–3665.

[72] Pulay, P. Analytical derivatives, forces, force constants, molecular geometries, and related response properties in electronic structure theory. *WIREs Comput Mol Sci* **2013**, *4*, 169–181.

[73] Hellmann, H. *Hans Hellmann: Einführung in die Quantenchemie*; Springer Berlin Heidelberg, 2015.

[74] Feynman, R. Forces in Molecules. *Phys. Rev.* **1939**, *56*, 340–343.

[75] Helgaker, T.; Jørgensen, P. Configuration-interaction energy derivatives in a fully variational formulation. *Theoret. Chim. Acta* **1989**, *75*, 111–127.

[76] Rehn, D. R.; Dreuw, A. Analytic nuclear gradients of the algebraic-diagrammatic construction scheme for the polarization propagator up to third order of perturbation theory. *J. Chem. Phys.* **2019**, *150*, 174110.

[77] Handy, N. C.; Schaefer, H. F. On the evaluation of analytic energy derivatives for correlated wave functions. *J. Chem. Phys.* **1984**, *81*, 5031–5033.

[78] Levchenko, S. V.; Wang, T.; Krylov, A. I. Analytic gradients for the spin-conserving and spin-flipping equation-of-motion coupled-cluster models with single and double substitutions. *J. Chem. Phys.* **2005**, *122*, 224106.

[79] Scheurer, M.; Reinholdt, P.; Kjellgren, E. R.; Haugaard Olsen, J. M.; Dreuw, A.; Kongsted, J. CPPE: An Open-Source C++ and Python Library for Polarizable Embedding. *J. Chem. Theory Comput.* **2019**, *15*, 6154–6163.

[80] Herbst, M. F.; Scheurer, M.; Fransson, T.; Rehn, D. R.; Dreuw, A. adcc: A versatile toolkit for rapid development of algebraic-diagrammatic construction methods. *WIREs Comput Mol Sci* **2020**, *10*, e1462.

[81] Parrish, R. M.; Burns, L. A.; Smith, D. G.; Simmonett, A. C.; DePrince, A. E.; Hohenstein, E. G.; Bozkaya, U.; Sokolov, A. Y.; Di Remigio, R.; Richard, R. M.; Gonthier, J. F.; James, A. M.; McAlexander, H. R.; Kumar, A.; Saitow, M.; Wang, X.; Pritchard, B. P.; Verma, P.; Schaefer, H. F.; Patkowski, K.; King, R. A.; Valeev, E. F.; Evangelista, F. A.; Turney, J. M.; Crawford, T. D.; Sherrill, C. D. Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability. *J. Chem. Theory Comput.* **2017**, *13*, 3185–3197.

[82] Smith, D. G.; Burns, L. A.; Sirianni, D. A.; Nascimento, D. R.; Kumar, A.; James, A. M.; Schriber, J. B.; Zhang, T.; Zhang, B.; Abbott, A. S.; Berquist, E. J.; Lechner, M. H.; Cunha, L. A.; Heide, A. G.; Waldrop, J. M.; Takeshita, T. Y.; Alenaizan, A.; Neuhauser, D.; King, R. A.; Simmonett, A. C.; Turney, J. M.; Schaefer, H. F.; Evangelista, F. A.; DePrince, A. E.; Crawford, T. D.; Patkowski, K.; Sherrill, C. D. Psi4NumPy: An Interactive Quantum Chemistry Programming Environment for Reference Implementations and Rapid Development. *J. Chem. Theory Comput.* **2018**, *14*, 3504–3511.

[83] Rinkevicius, Z.; Li, X.; Vahtras, O.; Ahmadzadeh, K.; Brand, M.; Ringholm, M.; List, N. H.; Scheurer, M.; Scott, M.; Dreuw, A.; Norman, P. VeloxChem: A Python-driven density-functional theory program for spectroscopy simulations in high-performance computing environments. *WIREs Comput Mol Sci* **2019**, *10*, e1457.

[84] Rehn, D. R.; Rinkevicius, Z.; Herbst, M. F.; Li, X.; Scheurer, M.; Brand, M.; Dempwolff, A. L.; Brumboiu, I. E.; Fransson, T.; Dreuw, A.; Norman, P. Gator: A Python-driven program for spectroscopy simulations using correlated wave functions. *WIREs Comput Mol Sci* **2021**, e1528.

[85] Kaliman, I. A.; Slipchenko, L. V. LIBEFP: A new parallel implementation of the effective fragment potential method as a portable software library. *J. Comput. Chem.* **2013**, *34*, 2284–2292.

[86] Di Remigio, R.; Steindal, A. H.; Mozgawa, K.; Weijo, V.; Cao, H.; Frediani, L. PCMSolver: An open-source library for solvation modeling. *Int J Quantum Chem* **2018**, *119*, e25685.

[87] Olsen, J. M. H.; List, N. H.; Steinmann, C.; Steindal, A. H.; Nørby, M. S.; Reinholdt, P. PElib: The Polarizable Embedding library. 2020; `https://doi.org/10.5281/zenodo.3967017`.

[88] Aidas, K.; Angeli, C.; Bak, K. L.; Bakken, V.; Bast, R.; Boman, L.; Christiansen, O.; Cimiraglia, R.; Coriani, S.; Dahle, P.; Dalskov, E. K.; Ekström, U.; Enevoldsen, T.; Eriksen, J. J.; Ettenhuber, P.; Fernández, B.; Ferrighi, L.; Fliegl, H.; Frediani, L.; Hald, K.; Halkier, A.; Hättig, C.; Heiberg, H.; Helgaker, T.; Hennum, A. C.; Hettema, H.; Hjertenaes, E.; Høst, S.; Høyvik, I.-M.; Iozzi, M. F.; Jansík, B.; Jensen, H. J. A.; Jonsson, D.; Jørgensen, P.; Kauczor, J.; Kirpekar, S.; Kjaergaard, T.; Klopper, W.; Knecht, S.; Kobayashi, R.; Koch, H.; Kongsted, J.; Krapp, A.; Kristensen, K.; Ligabue, A.; Lutnaes, O. B.; Melo, J. I.; Mikkelsen, K. V.; Myhre, R. H.; Neiss, C.; Nielsen, C. B.; Norman, P.; Olsen, J.; Olsen, J. M. H.; Osted, A.; Packer, M. J.; Pawlowski, F.; Pedersen, T. B.; Provasi, P. F.; Reine, S.; Rinkevicius, Z.; Ruden, T. A.; Ruud, K.; Rybkin, V. V.; Sałek, P.; Samson, C. C.; de Merás, A. S.; Saue, T.; Sauer, S. P.; Schimmelpfennig, B.; Sneskov, K.; Steindal, A. H.; Sylvester-Hvid, K. O.; Taylor, P. R.; Teale, A. M.; Tellgren, E. I.; Tew, D. P.; Thorvaldsen, A. J.; Thøgersen, L.; Vahtras, O.; Watson, M. A.; Wilson, D. J.; Ziolkowski, M.; Ågren, H. The Dalton quantum chemistry program system. *WIREs Comput Mol Sci* **2013**, *4*, 269–284.

[89] Hedegård, E. D.; List, N. H.; Jensen, H. J. A.; Kongsted, J. The multi-configuration self-consistent field method within a polarizable embedded framework. *J. Chem. Phys.* **2013**, *139*, 044101.

[90] Pedersen, M. N.; Hedegård, E. D.; Olsen, J. M. H.; Kauczor, J.; Norman, P.; Kongsted, J. Damped Response Theory in Combination with Polarizable Environments: The Polarizable Embedding Complex Polarization Propagator Method. *J. Chem. Theory Comput.* **2014**, *10*, 1164–1171.

[91] Hedegård, E. D.; Olsen, J. M. H.; Knecht, S.; Kongsted, J.; Jensen, H. J. A. Polarizable embedding with a multiconfiguration short-range density functional theory linear response method. *J. Chem. Phys.* **2015**, *142*, 114113.

[92] Hedegård, E. D.; Bast, R.; Kongsted, J.; Olsen, J. M. H.; Jensen, H. J. A. Relativistic Polarizable Embedding. *J. Chem. Theory Comput.* **2017**, *13*, 2870–2880.

[93] Shao, Y.; Gan, Z.; Epifanovsky, E.; Gilbert, A. T.; Wormit, M.; Kussmann, J.; Lange, A. W.; Behn, A.; Deng, J.; Feng, X.; Ghosh, D.; Goldey, M.; Horn, P. R.; Jacobson, L. D.; Kaliman, I.; Khaliullin, R. Z.; Kuś, T.; Landau, A.; Liu, J.; Proynov, E. I.; Rhee, Y. M.; Richard, R. M.; Rohrdanz, M. A.; Steele, R. P.; Sundstrom, E. J.; Woodcock, H. L.; Zimmerman, P. M.; Zuev, D.; Albrecht, B.; Alguire, E.; Austin, B.; Beran, G. J.; Bernard, Y. A.; Berquist, E.; Brandhorst, K.;

Bravaya, K. B.; Brown, S. T.; Casanova, D.; Chang, C.-M.; Chen, Y.; Chien, S. H.; Closser, K. D.; Crittenden, D. L.; Diedenhofen, M.; DiStasio, R. A.; Do, H.; Dutoi, A. D.; Edgar, R. G.; Fatehi, S.; Fusti-Molnar, L.; Ghysels, A.; Golubeva-Zadorozhnaya, A.; Gomes, J.; Hanson-Heine, M. W.; Harbach, P. H.; Hauser, A. W.; Hohenstein, E. G.; Holden, Z. C.; Jagau, T.-C.; Ji, H.; Kaduk, B.; Khistyaev, K.; Kim, J.; Kim, J.; King, R. A.; Klunzinger, P.; Kosenkov, D.; Kowalczyk, T.; Krauter, C. M.; Lao, K. U.; Laurent, A. D.; Lawler, K. V.; Levchenko, S. V.; Lin, C. Y.; Liu, F.; Livshits, E.; Lochan, R. C.; Luenser, A.; Manohar, P.; Manzer, S. F.; Mao, S.-P.; Mardirossian, N.; Marenich, A. V.; Maurer, S. A.; Mayhall, N. J.; Neuscamman, E.; Oana, C. M.; Olivares-Amaya, R.; O'Neill, D. P.; Parkhill, J. A.; Perrine, T. M.; Peverati, R.; Prociuk, A.; Rehn, D. R.; Rosta, E.; Russ, N. J.; Sharada, S. M.; Sharma, S.; Small, D. W.; Sodt, A.; Stein, T.; Stück, D.; Su, Y.-C.; Thom, A. J.; Tsuchimochi, T.; Vanovschi, V.; Vogt, L.; Vydrov, O.; Wang, T.; Watson, M. A.; Wenzel, J.; White, A.; Williams, C. F.; Yang, J.; Yeganeh, S.; Yost, S. R.; You, Z.-Q.; Zhang, I. Y.; Zhang, X.; Zhao, Y.; Brooks, B. R.; Chan, G. K.; Chipman, D. M.; Cramer, C. J.; Goddard, W. A.; Gordon, M. S.; Hehre, W. J.; Klamt, A.; Schaefer, H. F.; Schmidt, M. W.; Sherrill, C. D.; Truhlar, D. G.; Warshel, A.; Xu, X.; Aspuru-Guzik, A.; Baer, R.; Bell, A. T.; Besley, N. A.; Chai, J.-D.; Dreuw, A.; Dunietz, B. D.; Furlani, T. R.; Gwaltney, S. R.; Hsu, C.-P.; Jung, Y.; Kong, J.; Lambrecht, D. S.; Liang, W.; Ochsenfeld, C.; Rassolov, V. A.; Slipchenko, L. V.; Subotnik, J. E.; Van Voorhis, T.; Herbert, J. M.; Krylov, A. I.; Gill, P. M.; Head-Gordon, M. Advances in molecular quantum chemistry contained in the Q-Chem 4 program package. *Mol. Phys.* **2014**, *113*, 184–215.

[94] Epifanovsky, E.; Gilbert, A. T.; Feng, X.; Lee, J.; Mao, Y.; Mardirossian, N.; Pokhilko, P.; White, A. F.; Coons, M. P.; Dempwolff, A. L.; Gan, Z.; Hait, D.; Horn, P. R.; Jacobson, L. D.; Kaliman, I.; Kussmann, J.; Lange, A. W.; Lao, K. U.; Levine, D. S.; Liu, J.; McKenzie, S. C.; Morrison, A. F.; Nanda, K. D.; Plasser, F.; Rehn, D. R.; Vidal, M. L.; You, Z.-Q.; Zhu, Y.; Alam, B.; Albrecht, B. J.; Aldossary, A.; Alguire, E.; Andersen, J. H.; Athavale, V.; Barton, D.; Begam, K.; Behn, A.; Bellonzi, N.; Bernard, Y. A.; Berquist, E. J.; Burton, H. G. A.; Carreras, A.; Carter-Fenk, K.; Chakraborty, R.; Chien, A. D.; Closser, K. D.; Cofer-Shabica, V.; Dasgupta, S.; de Wergifosse, M.; Deng, J.; Diedenhofen, M.; Do, H.; Ehlert, S.; Fang, P.-T.; Fatehi, S.; Feng, Q.; Friedhoff, T.; Gayvert, J.; Ge, Q.; Gidofalvi, G.; Goldey, M.; Gomes, J.; González-Espinoza, C. E.; Gulania, S.; Gunina, A. O.; Hanson-Heine, M. W. D.; Harbach, P. H.; Hauser, A.; Herbst, M. F.; Hernández Vera, M.; Hodecker, M.; Holden, Z. C.; Houck, S.; Huang, X.; Hui, K.; Huynh, B. C.; Ivanov, M.; Jász, Á.; Ji, H.; Jiang, H.; Kaduk, B.; Kähler, S.; Khistyaev, K.; Kim, J.; Kis, G.; Klunzinger, P.; Koczor-Benda, Z.; Koh, J. H.; Kosenkov, D.; Koulias, L.; Kowalczyk, T.; Krauter, C. M.; Kue, K.; Kunitsa, A.; Kuś, T.; Ladjánszki, I.; Landau, A.; Lawler, K. V.; Lefrancois, D.; Lehtola, S.; Li, R. R.; Li, Y.-P.; Liang, J.; Liebenthal, M.; Lin, H.-H.; Lin, Y.-S.; Liu, F.; Liu, K.-Y.; Loipersberger, M.; Luenser, A.; Manjanath, A.; Manohar, P.; Mansoor, E.; Manzer, S. F.; Mao, S.-P.; Marenich, A. V.; Markovich, T.; Mason, S.; Maurer, S. A.; McLaughlin, P. F.; Menger, M. F.; Mewes, J.-M.; Mewes, S.; Morgante, P.; Mullinax, J. W.; Ooster-

baan, K. J.; Paran, G.; Paul, A. C.; Paul, S. K.; Pavošević, F.; Pei, Z.; Prager, S.; Proynov, E. I.; Rák, Á.; Ramos-Cordoba, E.; Rana, B.; Rask, A. E.; Rettig, A.; Richards, R. M.; Rob, F.; Rossomme, E.; Scheele, T.; Scheurer, M.; Schneider, M.; Sergueev, N.; Sharada, S. M.; Skomorowski, W.; Small, D. W.; Stein, C. J.; Su, Y.-C.; Sundstrom, E. J.; Tao, Z.; Thirman, J.; Tornai, G. J.; Tsuchimochi, T.; Tubman, N. M.; Veccham, S. P.; Vydrov, O.; Wenzel, J.; Witte, J.; Yamada, A.; Yao, K.; Yeganeh, S.; Yost, S. R.; Zech, A.; Zhang, I. Y.; Zhang, X.; Zhang, Y.; Zuev, D.; Aspuru-Guzik, A.; Bell, A. T.; Besley, N. A.; Bravaya, K. B.; Brooks, B. R.; Casanova, D.; Chai, J.-D.; Coriani, S.; Cramer, C.; Cserey, G.; DePrince III, A. E.; DiStasio Jr., R. A.; Dreuw, A.; Dunietz, B. D.; Furlani, T. R.; Goddard III, W. A.; Hammes-Schiffer, S.; Head-Gordon, T.; Hehre, W. J.; Hsu, C.-P.; Jagau, T.-C.; Jung, Y.; Klamt, A.; Kong, J.; Lambrecht, D. S.; Liang, W.; Mayhall, N. J.; McCurdy, C. W.; Neaton, J. B.; Ochsenfeld, C.; Parkhill, J. A.; Peverati, R.; Rassolov, V. A.; Shao, Y.; Slipchenko, L. V.; Stauch, T.; Steele, R. P.; Subotnik, J. E.; Thom, A. J. W.; Tkatchenko, A.; Truhlar, D. G.; Van Voorhis, T.; Wesolowski, T. A.; Whaley, K. B.; Woodcock III, H. L.; Zimmerman, P. M.; Faraji, S.; Gill, P. M.; Head-Gordon, M.; Herbert, J. M.; Krylov, A. I. Software for the frontiers of quantum chemistry: An overview of developments in the Q-Chem 5 package. *J. Chem. Phys.* **2021**,

[95] Sun, Q.; Berkelbach, T. C.; Blunt, N. S.; Booth, G. H.; Guo, S.; Li, Z.; Liu, J.; McClain, J. D.; Sayfutyarova, E. R.; Sharma, S.; Wouters, S.; Chan, G. K. P y SCF: The Python-based simulations of chemistry framework. *WIREs Comput Mol Sci* **2017**, *8*.

[96] CMake. https://cmake.org/.

[97] Guennebaud, G.; Jacob, B., et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[98] Jakob, W.; Rhinelander, J.; Moldovan, D. pybind11 – Seamless operability between C++11 and Python. 2017; https://github.com/pybind/pybind11.

[99] van der Walt, S.; Colbert, S. C.; Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Comput. Sci. Eng.* **2011**, *13*, 22–30.

[100] Krekel, H.; Oliveira, B.; Pfannschmidt, R.; Bruynooghe, F.; Laugher, B.; Bruhin, F. pytest. 2004; https://github.com/pytest-dev/pytest.

[101] Dykstra, C. E. Efficient calculation of electrically based intermolecular potentials of weakly bonded clusters. *J. Comput. Chem.* **1988**, *9*, 476–487.

[102] Meurer, A.; Smith, C. P.; Paprocki, M.; Čertík, O.; Kirpichev, S. B.; Rocklin, M.; Kumar, A.; Ivanov, S.; Moore, J. K.; Singh, S.; Rathnayake, T.; Vig, S.; Granger, B. E.; Muller, R. P.; Bonazzi, F.; Gupta, H.; Vats, S.; Johansson, F.; Pedregosa, F.; Curry, M. J.; Terrel, A. R.; Roučka, S.; Saboo, A.; Fernando, I.; Kulal, S.; Cimrman, R.; Scopatz, A. SymPy: Symbolic computing in Python. *PeerJ Comput. Sci.* **2017**, *3*, e103.

[103] Thole, B. Molecular polarizabilities calculated with a modified dipole interaction. *Chem. Phys.* **1981**, *59*, 341–350.

[104] van Duijnen, P. T.; Swart, M. Molecular and Atomic Polarizabilities: Thole's Model Revisited. *J. Phys. Chem. A* **1998**, *102*, 2399–2407.

[105] Reinholdt, P.; Kongsted, J.; Olsen, J. M. H. Polarizable Density Embedding: A Solution to the Electron Spill-Out Problem in Multiscale Modeling. *J. Phys. Chem. Lett.* **2017**, *8*, 5949–5958.

[106] Marefat Khah, A.; Reinholdt, P.; Olsen, J. M. H.; Kongsted, J.; Hättig, C. Avoiding Electron Spill-Out in QM/MM Calculations on Excited States with Simple Pseudopotentials. *J. Chem. Theory Comput.* **2020**, *16*, 1373–1381.

[107] Banerjee, S.; Sokolov, A. Y. Third-order algebraic diagrammatic construction theory for electron attachment and ionization energies: Conventional and Green's function implementation. *J. Chem. Phys.* **2019**, *151*, 224112.

[108] Epifanovsky, E.; Wormit, M.; Kuś, T.; Landau, A.; Zuev, D.; Khistyaev, K.; Manohar, P.; Kaliman, I.; Dreuw, A.; Krylov, A. I. New implementation of high-level correlated methods using a general block tensor library for high-performance electronic structure calculations. *J. Comput. Chem.* **2013**, *34*, 2293–2309.

[109] Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95.

[110] Herbst, M. F.; Dreuw, A.; Avery, J. E. Toward quantum-chemical method development for arbitrary basis functions. *J. Chem. Phys.* **2018**, *149*, 084106.

[111] Herbst, M. F. Development of a modular quantum-chemistry framework for the investigation of novel basis functions. Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg, 2018.

[112] McKinney, W. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference. 2010; pp 51–56.

[113] McKinney, W. pandas: A Foundational Python Library for Data Analysis and Statistics. **2011**,

[114] G. A. Smith, D.; Gray, J. opt_einsum - A Python package for optimizing contraction order for einsum-like expressions. *JOSS* **2018**, *3*, 753.

[115] Mewes, J.-M.; You, Z.-Q.; Wormit, M.; Kriesche, T.; Herbert, J. M.; Dreuw, A. Experimental Benchmark Data and Systematic Evaluation of Two a Posteriori, Polarizable-Continuum Corrections for Vertical Excitation Energies in Solution. *J. Phys. Chem. A* **2015**, *119*, 5446–5464.

[116] Mewes, J.-M.; Herbert, J. M.; Dreuw, A. On the accuracy of the general, state-specific polarizable-continuum model for the description of correlated ground- and excited states in solution. *Phys. Chem. Chem. Phys.* **2017**, *19*, 1644–1654.

[117] Grininger, M.; Staudt, H.; Johansson, P.; Wachtveitl, J.; Oesterhelt, D. Dodecin Is the Key Player in Flavin Homeostasis of Archaea. *J. Biol. Chem.* **2009**, *284*, 13068–13076.

[118] Staudt, H.; Oesterhelt, D.; Grininger, M.; Wachtveitl, J. Ultrafast Excited-state Deactivation of Flavins Bound to Dodecin. *J. Biol. Chem.* **2012**, *287*, 17637–17644.

[119] Staudt, H.; Hoesl, M. G.; Dreuw, A.; Serdjukow, S.; Oesterhelt, D.; Budisa, N.; Wachtveitl, J.; Grininger, M. Directed Manipulation of a Flavoprotein Photocycle. *Angew. Chem. Int. Ed.* **2013**, *52*, 8463–8466.

[120] Scheurer, M.; Brisker-Klaiman, D.; Dreuw, A. Molecular Mechanism of Flavin Photoprotection by Archaeal Dodecin: Photoinduced Electron Transfer and Mg2+-Promoted Proton Transfer. *J. Phys. Chem. B* **2017**, *121*, 10457–10466.

[121] Grabarek, D.; Andruniów, T. Removing artifacts in polarizable embedding calculations of one- and two-photon absorption spectra of fluorescent proteins. *J. Chem. Phys.* **2020**, *153*, 215102.

[122] List, N. H.; Jensen, H. J. A.; Kongsted, J. Local electric fields and molecular properties in heterogeneous environments through polarizable embedding. *Phys. Chem. Chem. Phys.* **2016**, *18*, 10070–10080.

[123] Zuev, D.; Bravaya, K. B.; Makarova, M. V.; Krylov, A. I. Effect of microhydration on the electronic structure of the chromophores of the photoactive yellow and green fluorescent proteins. *J. Chem. Phys.* **2011**, *135*, 194304.

[124] Gagliardi, L.; Lindh, R.; Karlström, G. Local properties of quantum chemical systems: The LoProp approach. *J. Chem. Phys.* **2004**, *121*, 4494–4500.

[125] Olsen, J. M. H. PyFraME: Python framework for Fragment-based Multiscale Embedding. 2020; `https://doi.org/10.5281/zenodo.3820471`.

[126] Vahtras, O. LoProp for Dalton. 2014.

[127] Yanai, T.; Tew, D. P.; Handy, N. C. A new hybrid exchange–correlation functional using the Coulomb-attenuating method (CAM-B3LYP). *Chem. Phys. Lett.* **2004**, *393*, 51–57.

[128] Dunning, T. H. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *J. Chem. Phys.* **1989**, *90*, 1007–1023.

[129] Seritan, S.; Bannwarth, C.; Fales, B. S.; Hohenstein, E. G.; Isborn, C. M.; Kokkila-Schumacher, S. I.; Li, X.; Liu, F.; Luehr, N.; Snyder, J. W.; Song, C.; Titov, A. V.; Ufimtsev, I. S.; Wang, L.; Martínez, T. J. TeraChem : A graphical processing unit -accelerated electronic structure package for large-scale ab initio molecular dynamics. *WIREs Comput Mol Sci* **2020**, *11*, e1494.

[130] List, N. H.; Olsen, J. M. H.; Jensen, H. J. A.; Steindal, A. H.; Kongsted, J. Molecular-Level Insight into the Spectral Tuning Mechanism of the DsRed Chromophore. *J. Phys. Chem. Lett.* **2012**, *3*, 3513–3521.

[131] Adamo, C.; Barone, V. Toward reliable density functional methods without adjustable parameters: The PBE0 model. *J. Chem. Phys.* **1999**, *110*, 6158–6170.

[132] Balasubramani, S. G.; Chen, G. P.; Coriani, S.; Diedenhofen, M.; Frank, M. S.; Franzke, Y. J.; Furche, F.; Grotjahn, R.; Harding, M. E.; Hättig, C.; Hellweg, A.; Helmich-Paris, B.; Holzer, C.; Huniar, U.; Kaupp, M.; Marefat Khah, A.; Karbalaei Khani, S.; Müller, T.; Mack, F.; Nguyen, B. D.; Parker, S. M.; Perlt, E.; Rappoport, D.; Reiter, K.; Roy, S.; Rückert, M.; Schmitz, G.; Sierka, M.; Tapavicza, E.; Tew, D. P.; van Wüllen, C.; Voora, V. K.; Weigend, F.; Wodyński, A.; Yu, J. M. TURBOMOLE: Modular program suite for ab initio quantum-chemical and condensed-matter simulations. *J. Chem. Phys.* **2020**, *152*, 184107.

[133] Weigend, F.; Köhn, A.; Hättig, C. Efficient use of the correlation consistent basis sets in resolution of the identity MP2 calculations. *J. Chem. Phys.* **2002**, *116*, 3175–3183.

[134] Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **1987**, *73*, 325–348.

[135] Coles, J. P.; Masella, M. The fast multipole method and point dipole moment polarizable force fields. *J. Chem. Phys.* **2015**, *142*, 024109.

[136] Lipparini, F. General Linear Scaling Implementation of Polarizable Embedding Schemes. *J. Chem. Theory Comput.* **2019**, *15*, 4312–4317.

[137] Dehnen, W. A Very Fast and Momentum-conserving Tree Code. *Astrophys. J.* **2000**, *536*, L39–L42.

[138] Visscher, P.; Apalkov, D. Simple recursive implementation of fast multipole method. *J. Magn. Magn. Mater.* **2010**, *322*, 275–281.

[139] Pepper, R. A.; Fangohr, H. fmmgen: Automatic Code Generation of Operators for Cartesian Fast Multipole and Barnes-Hut Methods. **2020**,

[140] Scheurer, M.; Reinholdt, P.; Olsen, J. M. H.; Dreuw, A.; Kongsted, J. Efficient Open-Source Implementations of Linear-Scaling Polarizable Embedding: Use Octrees to Save the Trees. *J. Chem. Theory Comput.* **2021**, *17*, 3445–3454.

[141] Saue, T.; Bast, R.; Gomes, A. S. P.; Jensen, H. J. A.; Visscher, L.; Aucar, I. A.; Di Remigio, R.; Dyall, K. G.; Eliav, E.; Fasshauer, E., et al. The DIRAC code for relativistic molecular calculations. *J. Chem. Phys.* **2020**, *152*, 204104.

[142] Martínez, L.; Andrade, R.; Birgin, E.; Martínez, J. PACKMOL: A package for building initial configurations for molecular dynamics simulations. *J. Comput. Chem.* **2009**, *30*, 2157–2164.

[143] Hehre, W.; Ditchfield, R.; Pople, J. Self—Consistent Molecular Orbital Methods. XII. Further Extensions of Gaussian—Type Basis Sets for Use in Molecular Orbital Studies of Organic Molecules. *J. Chem. Phys.* **1972**, *56*, 2257–2261.

[144] Hirata, S.; Head-Gordon, M. Time-dependent density functional theory within the Tamm–Dancoff approximation. *Chem. Phys. Lett.* **1999**, *314*, 291–299.

[145] Reinholdt, P.; Kjellgren, E. R.; Steinmann, C.; Olsen, J. M. H. Cost-Effective Potential for Accurate Polarizable Embedding Calculations in Protein Environments. *J. Chem. Theory Comput.* **2019**, *16*, 1162–1174.

[146] Beerepoot, M. T.; Steindal, A. H.; List, N. H.; Kongsted, J.; Olsen, J. M. H. Averaged Solvent Embedding Potential Parameters for Multiscale Modeling of Molecular Properties. *J. Chem. Theory Comput.* **2016**, *12*, 1684–1695.

[147] Caprasecca, S.; Jurinovich, S.; Lagardère, L.; Stamm, B.; Lipparini, F. Achieving Linear Scaling in Computational Cost for a Fully Polarizable MM/Continuum Embedding. *J. Chem. Theory Comput.* **2015**, *11*, 694–704.

[148] Loos, P.-F.; Scemama, A.; Blondel, A.; Garniron, Y.; Caffarel, M.; Jacquemin, D. A Mountaineering Strategy to Excited States: Highly Accurate Reference Energies and Benchmarks. *J. Chem. Theory Comput.* **2018**, *14*, 4360–4379.

[149] Suellen, C.; Freitas, R. G.; Loos, P.-F.; Jacquemin, D. Cross-Comparisons between Experiment, TD-DFT, CC, and ADC for Transition Energies. *J. Chem. Theory Comput.* **2019**, *15*, 4581–4590.

[150] Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. Graphics* **1996**, *14*, 33–38.

[151] Kauczor, J.; Jørgensen, P.; Norman, P. On the Efficiency of Algorithms for Solving Hartree–Fock and Kohn–Sham Response Equations. *J. Chem. Theory Comput.* **2011**, *7*, 1610–1630.

[152] Hestenes, M.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. RES. NATL. BUR. STAN.* **1952**, *49*, 409.

[153] Nanda, K. D.; Vidal, M. L.; Faber, R.; Coriani, S.; Krylov, A. I. How to stay out of trouble in RIXS calculations within equation-of-motion coupled-cluster damped response theory? Safe hitchhiking in the excitation manifold by means of core–valence separation. *Phys. Chem. Chem. Phys.* **2020**, *22*, 2629–2641.

[154] Jonsson, D.; Norman, P.; Luo, Y.; Ågren, H. Response theory for static and dynamic polarizabilities of excited states. *J. Chem. Phys.* **1996**, *105*, 581–587.

[155] Hättig, C.; Christiansen, O.; Coriani, S.; Jo/rgensen, P. Static and frequency-dependent polarizabilities of excited singlet states using coupled cluster response theory. *J. Chem. Phys.* **1998**, *109*, 9237–9243.

[156] Nanda, K. D.; Krylov, A. I. Static polarizabilities for excited states within the spin-conserving and spin-flipping equation-of-motion coupled-cluster singles and doubles formalism: Theory, implementation, and benchmarks. *J. Chem. Phys.* **2016**, *145*, 204116.

[157] Norman, P.; Jonsson, D.; Ågren, H. Excited state properties through cubic response theory: Polarizabilities of benzene and naphthalene. *Chem. Phys. Lett.* **1997**, *268*, 337–344.

[158] Jonsson, D.; Norman, P.; Ågren, H. Single determinant calculations of excited state polarizabilities. *Chem. Phys.* **1997**, *224*, 201–214.

[159] Jonsson, D.; Norman, P.; Ågren, H.; Luo, Y.; Sylvester-Hvid, K. O.; Mikkelsen, K. V. Excited state polarizabilities in solution obtained by cubic response theory: Calculations on para-, ortho-, and meta-nitroaniline. *J. Chem. Phys.* **1998**, *109*, 6351–6357.

[160] Stanton, J. F.; Gauss, J. The first excited singlet state of s-tetrazine: A theoretical analysis of some outstanding questions. *J. Chem. Phys.* **1996**, *104*, 9859–9869.

[161] Pašteka, L. F.; Melicherčík, M.; Neogrády, P.; Urban, M. CASPT2 and CCSD(T) calculations of dipole moments and polarizabilities of acetone in excited states. *Mol. Phys.* **2012**, *110*, 2219–2237.

[162] Graf, N. K.; Friese, D. H.; Winter, N. O.; Hättig, C. Excited state polarizabilities for CC2 using the resolution-of-the-identity approximation. *J. Chem. Phys.* **2015**, *143*, 244108.

[163] Medveď, M.; Budzák, S.; Pluta, T. Electric properties of the low-lying excited states of benzonitrile: Geometry relaxation and solvent effects. *Theor Chem Acc* **2015**, *134*, 78.

[164] Schütz, M.; Hutter, J.; Lüthi, H. P. The molecular and electronic structure ofs-tetrazine in the ground and first excited state: A theoretical investigation. *J. Chem. Phys.* **1995**, *103*, 7048–7057.

[165] Jansik, B.; Jonsson, D.; Sałek, P.; Ågren, H. Calculations of static and dynamic polarizabilities of excited states by means of density functional theory. *J. Chem. Phys.* **2004**, *121*, 7595.

[166] Norman, P.; Jiemchooroj, A.; Sernelius, B. E. First principle calculations of dipole-dipole dispersion coefficients for the ground and first $\pi \to \pi^*$ excited states of some azabenzenes. *JCM* **2004**, *4*, 321–332.

[167] Pauszek, R. F.; Kodali, G.; Stanley, R. J. Excited State Electronic Structures of 5,10-Methenyltetrahydrofolate and 5,10-Methylenetetrahydrofolate Determined by Stark Spectroscopy. *J. Phys. Chem. A* **2014**, *118*, 8320–8328.

[168] Stanton, J. F.; Bartlett, R. J. The equation of motion coupled-cluster method. A systematic biorthogonal approach to molecular excitation energies, transition probabilities, and excited state properties. *J. Chem. Phys.* **1993**, *98*, 7029–7039.

[169] Nanda, K. D.; Krylov, A. I. Two-photon absorption cross sections within equation-of-motion coupled-cluster formalism using resolution-of-the-identity and Cholesky decomposition representations: Theory, implementation, and benchmarks. *J. Chem. Phys.* **2015**, *142*, 064118.

[170] Nanda, K. D.; Krylov, A. I. Effect of the diradical character on static polarizabilities and two-photon absorption cross sections: A closer look with spin-flip equation-of-motion coupled-cluster singles and doubles method. *J. Chem. Phys.* **2017**, *146*, 224103.

[171] Sadlej, A. J. Medium-size polarized basis sets for high-level correlated calculations of molecular electric properties. *Collect. Czech. Chem. Commun.* **1988**, *53*, 1995–2016.

[172] O'boyle, N. M.; Tenderholt, A. L.; Langner, K. M. cclib: A library for package-independent computational chemistry algorithms. *J. Comput. Chem.* **2008**, *29*, 839–845.

[173] Waskom, M.; Botvinnik, O.; Ostblom, J.; Lukauskas, S.; Hobson, P.; MaozGelbart,; Gemperline, D. C.; Augspurger, T.; Halchenko, Y.; Cole, J. B.; Warmenhoven, J.; de Ruiter, J.; Pye, C.; Hoyer, S.; Vanderplas, J.; Villalba, S.; Kunter, G.; Quintero, E.; Bachant, P.; Martin, M.; Meyer, K.; Swain, C.; Miles, A.; Brunner, T.; O'Kane, D.; Yarkoni, T.; Williams, M. L.; Evans, C. mwaskom/seaborn: V0.10.0. 2020; `https://doi.org/10.5281/zenodo.3629446`.

[174] Heitz, S.; Weidauer, D.; Hese, A. Measurement of static polarizabilities on s-tetrazine. *J. Chem. Phys.* **1991**, *95*, 7952–7956.

[175] Hurdis, E. C.; Smyth, C. P. The Structural Effects of Unsaturation and Hyperconjugation in Aldehydes, Nitriles and Chlorides as Shown by their Dipole Moments in the Vapor State. *J. Am. Chem. Soc.* **1943**, *65*, 89–96.

[176] Heitz, S.; Weidauer, D.; Rosenow, B.; Hese, A. Measurement of static polarizabilities on C10H8 and C10D8. *J. Chem. Phys.* **1992**, *96*, 976–981.

[177] Dreuw, A.; Head-Gordon, M. Failure of Time-Dependent Density Functional Theory for Long-Range Charge-Transfer Excited States: The Zincbacteriochlorin−Bacteriochlorin and Bacteriochlorophyll−Spheroidene Complexes. *J. Am. Chem. Soc.* **2004**, *126*, 4007–4016.

[178] Eriksen, J. J.; Sauer, S. P.; Mikkelsen, K. V.; Christiansen, O.; Jensen, H. J. A.; Kongsted, J. Failures of TDDFT in describing the lowest intramolecular charge-transfer excitation in para-nitroaniline. *Mol. Phys.* **2013**, *111*, 1235–1248.

[179] Casimir, H.; Polder, D. The Influence of Retardation on the London-van der Waals Forces. *Phys. Rev.* **1948**, *73*, 360–372.

[180] Lefrancois, D.; Wormit, M.; Dreuw, A. Adapting algebraic diagrammatic construction schemes for the polarization propagator to problems with multi-reference electronic ground states exploiting the spin-flip ansatz. *J. Chem. Phys.* **2015**, *143*, 124107.

[181] Scheurer, M.; Dreuw, A.; Epifanovsky, E.; Head-Gordon, M.; Stauch, T. Modeling Molecules under Pressure with Gaussian Potentials. *J. Chem. Theory Comput.* **2020**, *17*, 583–597.

[182] Scheurer, M.; Dreuw, A.; Head-Gordon, M.; Stauch, T. The rupture mechanism of rubredoxin is more complex than previously thought. *Chem. Sci.* **2020**, *11*, 6036–6044.

[183] Grochala, W.; Hoffmann, R.; Feng, J.; Ashcroft, N. W. The Chemical Imagination at Work inVery Tight Places. *Angew. Chem. Int. Ed.* **2007**, *46*, 3620–3642.

[184] Schettino, V.; Bini, R. Constraining molecules at the closest approach: Chemistry at high pressure. *Chem. Soc. Rev.* **2007**, *36*, 869.

[185] McMillan, P. F. Chemistry at high pressure. *Chem. Soc. Rev.* **2006**, *35*, 855.

[186] Li, B.; Ji, C.; Yang, W.; Wang, J.; Yang, K.; Xu, R.; Liu, W.; Cai, Z.; Chen, J.; Mao, H.-k. Diamond anvil cell behavior up to 4 Mbar. *Proc Natl Acad Sci USA* **2018**, *115*, 1713–1717.

[187] Asay, J. R., Shahinpoor, M., Eds. *High-Pressure Shock Compression of Solids*, 1st ed.; Springer New York: New York, 1993.

[188] Stauch, T. Quantum chemical modeling of molecules under pressure. *Int J Quantum Chem* **2020**, *121*, e26208.

[189] Pašteka, L.; Helgaker, T.; Saue, T.; Sundholm, D.; Werner, H.-J.; Hasanbulli, M.; Major, J.; Schwerdtfeger, P. Atoms and molecules in soft confinement potentials. *Mol. Phys.* **2020**, *118*, e1730989.

[190] Novoa, T.; Contreras-García, J.; Fuentealba, P.; Cárdenas, C. The Pauli principle and the confinement of electron pairs in a double well: Aspects of electronic bonding under pressure. *J. Chem. Phys.* **2019**, *150*, 204304.

[191] Aquino, N. The Hydrogen and Helium Atoms Confined in Spherical Boxes. *Adv. Quantum Chem.* **2009**, *57*, 123–171.

[192] Gorecki, J.; Byers Brown, W. On the ground state of the hydrogen molecule–ion H+2enclosed in hard and soft spherical boxes. *J. Chem. Phys.* **1988**, *89*, 2138–2148.

[193] Weisskopf, V. F. Search for Simplicity: Quantum mechanics and the Pauli principle. *Am. J. Phys.* **1985**, *53*, 109–110.

[194] LeSar, R.; Herschbach, D. Electronic and vibrational properties of molecules at high pressures. Hydrogen molecule in a rigid spheroidal box. *J. Phys. Chem.* **1981**, *85*, 2798–2804.

[195] Ley-Koo, E.; Rubinstein, S. The hydrogen atom within spherical boxes with penetrable walls. *J. Chem. Phys.* **1979**, *71*, 351–357.

[196] Ludeña, E. V. SCF calculations for hydrogen in a spherical box. *J. Chem. Phys.* **1977**, *66*, 468–470.

[197] Suryanarayana, D.; Weil, J. On the hyperfine splitting of the hydrogen atom in a spherical box. *J. Chem. Phys.* **1976**, *64*, 510–513.

[198] Michels, A.; De Boer, J.; Bijl, A. Remarks concerning molecural interaction and their influence on the polarisability. *Physica* **1937**, *4*, 981–994.

[199] Zurek, E.; Grochala, W. Predicting crystal structures and properties of matter under extreme conditions via quantum mechanics: The pressure is on. *Phys. Chem. Chem. Phys.* **2015**, *17*, 2917–2934.

[200] Subramanian, G.; Mathew, N.; Leiding, J. A generalized force-modified potential energy surface for mechanochemical simulations. *J. Chem. Phys.* **2015**, *143*, 134109.

[201] Ong, M. T.; Leiding, J.; Tao, H.; Virshup, A. M.; Martínez, T. J. First Principles Dynamics and Minimum Energy Pathways for Mechanochemical Ring Opening of Cyclobutene. *J. Am. Chem. Soc.* **2009**, *131*, 6377–6379.

[202] Stauch, T.; Chakraborty, R.; Head-Gordon, M. Quantum Chemical Modeling of Pressure-Induced Spin Crossover in Octahedral Metal-Ligand Complexes. *ChemPhysChem* **2019**, *20*, 2742–2747.

[203] Stauch, T. A mechanochemical model for the simulation of molecules and molecular crystals under hydrostatic pressure. *J. Chem. Phys.* **2020**, *153*, 134503.

[204] Cammi, R.; Verdolino, V.; Mennucci, B.; Tomasi, J. Towards the elaboration of a QM method to describe molecular solutes under the effect of a very high pressure. *Chem. Phys.* **2008**, *344*, 135–141.

[205] Cammi, R. A new extension of the polarizable continuum model: Toward a quantum chemical description of chemical reactions at extreme high pressure. *J. Comput. Chem.* **2015**, *36*, 2246–2259.

[206] Chen, B.; Hoffmann, R.; Cammi, R. The Effect of Pressure on Organic Reactions in Fluids—a New Theoretical Perspective. *Angew. Chem. Int. Ed.* **2017**, *56*, 11126–11142.

[207] Fukuda, R.; Nakatani, K. Quantum Chemical Study on the High-Pressure Effect for [4 + 4] Retrocycloaddition of Anthracene Cyclophane Photodimer. *J. Phys. Chem. C* **2019**, *123*, 4493–4501.

[208] Pagliai, M.; Cardini, G.; Cammi, R. Vibrational Frequencies of Fullerenes C60and C70under Pressure Studied with a Quantum Chemical Model Including Spatial Confinement Effects. *J. Phys. Chem. A* **2014**, *118*, 5098–5111.

[209] Fukuda, R.; Ehara, M.; Cammi, R. Modeling Molecular Systems at Extreme Pressure by an Extension of the Polarizable Continuum Model (PCM) Based on the

Symmetry-Adapted Cluster-Configuration Interaction (SAC–CI) Method: Confined Electronic Excited States of Furan as a Test Case. *J. Chem. Theory Comput.* **2015**, *11*, 2063–2076.

[210] Rahm, M.; Cammi, R.; Ashcroft, N.; Hoffmann, R. Squeezing All Elements in the Periodic Table: Electron Configuration and Electronegativity of the Atoms under Compression. *J. Am. Chem. Soc.* **2019**, *141*, 10253–10271.

[211] Caratelli, C.; Cammi, R.; Chelli, R.; Pagliai, M.; Cardini, G.; Schettino, V. Insights on the Realgar Crystal Under Pressure from XP-PCM and Periodic Model Calculations. *J. Phys. Chem. A* **2017**, *121*, 8825–8834.

[212] Cammi, R.; Cappelli, C.; Mennucci, B.; Tomasi, J. Calculation and analysis of the harmonic vibrational frequencies in molecules at extreme pressure: Methodology and diborane as a test case. *J. Chem. Phys.* **2012**, *137*, 154112.

[213] Hohenberg, P.; Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **1964**, *136*, B864–B871.

[214] Kohn, W.; Sham, L. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, A1133–A1138.

[215] Lange, A. W.; Herbert, J. M. Polarizable Continuum Reaction-Field Solvation Models Affording Smooth Potential Energy Surfaces. *J. Phys. Chem. Lett.* **2009**, *1*, 556–561.

[216] Rahm, M.; Ångqvist, M.; Rahm, J. M.; Erhart, P.; Cammi, R. Non-Bonded Radii of the Atoms Under Compression. *ChemPhysChem* **2020**, *21*, 2441–2453.

[217] Obara, S.; Saika, A. Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1986**, *84*, 3963–3974.

[218] Pulay, P. Ab initio calculation of force constants and equilibrium geometries in polyatomic molecules. *Mol. Phys.* **1969**, *17*, 197–204.

[219] Cossi, M.; Mennucci, B.; Cammi, R. Analytical first derivatives of molecular surfaces with respect to nuclear coordinates. *J. Comput. Chem.* **1996**, *17*, 57–73.

[220] Cossi, M.; Scalmani, G.; Rega, N.; Barone, V. New developments in the polarizable continuum model for quantum mechanical and classical calculations on molecules in solution. *J. Chem. Phys.* **2002**, *117*, 43–54.

[221] Wolfram, Mathematica, Version 12.0. `https://www.wolfram.com/mathematica`.

[222] Bondi, A. van der Waals Volumes and Radii. *J. Phys. Chem.* **1964**, *68*, 441–451.

[223] Wong, M. W.; Wiberg, K. B.; Frisch, M. J. Ab initio calculation of molar volumes: Comparison with experiment and use in solvation models. *J. Comput. Chem.* **1995**, *16*, 385–394.

[224] Jha, S. K.; Brown, K.; Todde, G.; Subramanian, G. A mechanochemical study of the effects of compression on a Diels-Alder reaction. *J. Chem. Phys.* **2016**, *145*, 074307.

[225] Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865–3868.

[226] Vollhardt, K. P. C. Transition-metal-catalyzed acetylene cyclizations in organic synthesis. *Acc. Chem. Res.* **1977**, *10*, 1–8.

[227] Breitkopf, V.; Hopf, H.; Klärner, F.-G.; Witulski, B.; Zimny, B. The effect of pressure on the trimerization and Diels-Alder reaction of cyanoacetylene. Synthesis and reactivity of 2,3,5-tricyanobicyclo[2.2.0]hexa-2,5-diene („tricyano Dewar benzene"). *Liebigs Ann.* **1995**, *1995*, 613–617.

[228] Becke, A. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A* **1988**, *38*, 3098–3100.

[229] Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B* **1988**, *37*, 785–789.

[230] Becke, A. D. A new mixing of Hartree–Fock and local density-functional theories. *J. Chem. Phys.* **1993**, *98*, 1372–1377.

[231] Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. *J. Chem. Phys.* **2010**, *132*, 154104.

[232] Robertson, W.; Babb, S.; Matsen, F. Effects of High Pressure on the Near Ultraviolet Absorption Spectra of Benzene and Monosubstituted Benzenes. *J. Chem. Phys.* **1957**, *26*, 367–370.

[233] Aust, R.; Bentley, W.; Drickamer, H. Behavior of Fused-Ring Aromatic Hydrocarbons at Very High Pressure. *J. Chem. Phys.* **1964**, *41*, 1856–1864.

[234] Neumann, B.; Pollmann, P. Investigation of two cyanine dyes at normal and high pressure by UV/Vis spectroscopy. *Phys. Chem. Chem. Phys.* **2000**, *2*, 4784–4792.

[235] Puschnig, P.; Ambrosch-Draxl, C.; Heimel, G.; Zojer, E.; Resel, R.; Leising, G.; Kriechbaum, M.; Graupner, W. Pressure studies on the intermolecular interactions in biphenyl. *Synthetic Met.* **2001**, *116*, 327–331.

[236] Schönfelder, J.; Alonso-Caballero, A.; De Sancho, D.; Perez-Jimenez, R. The life of proteins under mechanical force. *Chem. Soc. Rev.* **2018**, *47*, 3558–3573.

[237] Stauch, T.; Dreuw, A. Advances in Quantum Mechanochemistry: Electronic Structure Methods and Force Analysis. *Chem. Rev.* **2016**, *116*, 14137–14180.

[238] Verdorfer, T.; Bernardi, R. C.; Meinhold, A.; Ott, W.; Luthey-Schulten, Z.; Nash, M. A.; Gaub, H. E. Combiningin Vitroandin SilicoSingle-Molecule Force Spectroscopy to Characterize and Tune Cellulosomal Scaffoldin Mechanics. *J. Am. Chem. Soc.* **2017**, *139*, 17841–17852.

[239] Milles, L. F.; Schulten, K.; Gaub, H. E.; Bernardi, R. C. Molecular mechanism of extreme mechanostability in a pathogen adhesin. *Science* **2018**, *359*, 1527–1533.

[240] Garcia-Manyes, S.; Beedle, A. E. Steering chemical reactions with force. *Nat Rev Chem* **2017**, *1*, DOI: 10.1038/s41570–017–0083.

[241] Li, J.; Li, H. Mechanical Unfolding Pathway of the High-Potential Iron–Sulfur Protein Revealed by Single-Molecule Atomic Force Microscopy: Toward a General Unfolding Mechanism for Iron–sulfur Proteins. *J. Phys. Chem. B* **2018**, *122*, 9340–9349.

[242] Johnson, D. C.; Dean, D. R.; Smith, A. D.; Johnson, M. K. Structure, Function, And Formation Of Biological Iron-Sulfur Clusters. *Annu. Rev. Biochem.* **2005**, *74*, 247–281.

[243] Beinert, H. Iron-sulfur proteins: Ancient structures, still full of surprises. *J Biol Inorg Chem* **2000**, *5*, 2–15.

[244] Holm, R. H.; Kennepohl, P.; Solomon, E. I. Structural and Functional Aspects of Metal Sites in Biology. *Chem. Rev.* **1996**, *96*, 2239–2314.

[245] Solomon, E. I.; Gorelsky, S. I.; Dey, A. Metal–thiolate bonds in bioinorganic chemistry. *J. Comput. Chem.* **2006**, *27*, 1415–1428.

[246] Zheng, P.; Li, H. Highly Covalent Ferric−Thiolate Bonds Exhibit Surprisingly Low Mechanical Stability. *J. Am. Chem. Soc.* **2011**, *133*, 6791–6798.

[247] Beyer, M. K.; Clausen-Schaumann, H. Mechanochemistry: The Mechanical Activation of Covalent Bonds. *Chem. Rev.* **2005**, *105*, 2921–2948.

[248] Arantes, G. M.; Bhattacharjee, A.; Field, M. J. Homolytic Cleavage of Fe-S Bonds in Rubredoxin under Mechanical Stress. *Angew. Chem. Int. Ed.* **2013**, *52*, 8144–8146.

[249] Arantes, G. M.; Field, M. J. Ferric–Thiolate Bond Dissociation Studied with Electronic Structure Calculations. *J. Phys. Chem. A* **2015**, *119*, 10084–10090.

[250] Zheng, P.; Arantes, G. M.; Field, M. J.; Li, H. Force-induced chemical reactions on the metal centre in a single metalloprotein molecule. *Nat Commun* **2015**, *6*, 7569.

[251] Zheng, P.; Chou, C.-C.; Guo, Y.; Wang, Y.; Li, H. Single Molecule Force Spectroscopy Reveals the Molecular Mechanical Anisotropy of the FeS4 Metal Center in Rubredoxin. *J. Am. Chem. Soc.* **2013**, *135*, 17783–17792.

[252] Nunes-Alves, A.; Arantes, G. M. Mechanical Unfolding of Macromolecules Coupled to Bond Dissociation. *J. Chem. Theory Comput.* **2017**, *14*, 282–290.

[253] Kurihara, K.; Tanaka, I.; Chatake, T.; Adams, M.; Jenney, F.; Moiseeva, N.; Bau, R.; Niimura, N. Neutron crystallographic study on rubredoxin from Pyrococcus furiosus by BIX-3, a single-crystal diffractometer for biomacromolecules. *Proc. Natl. Acad. Sci.* **2004**, *101*, 11215–11220.

[254] Zheng, P.; Takayama, S.-i. J.; Mauk, A. G.; Li, H. Hydrogen Bond Strength Modulates the Mechanical Strength of Ferric-Thiolate Bonds in Rubredoxin. *J. Am. Chem. Soc.* **2012**, *134*, 4124–4131.

[255] Lin, I.-J.; Gebel, E.; Machonkin, T.; Westler, W.; Markley, J. Changes in hydrogen-bond strengths explain reduction potentials in 10 rubredoxin variants. *Proc. Natl. Acad. Sci.* **2005**, *102*, 14581–14586.

[256] Okamura, T.-a.; Takamizawa, S.; Ueyama, N.; Nakamura, A. Novel Rubredoxin Model Tetrathiolato Iron(II) and Cobalt(II) Complexes Containing Intramolecular Single and Double NH···S Hydrogen Bonds. *Inorg. Chem.* **1998**, *37*, 18–28.

[257] Bau, R.; Rees, D.; Kurtz, D. M.; Scott, R. A.; Huang, H.; Adams, M. W.; Eidsness, M. Crystal structure of rubredoxin from Pyrococcus furiosus at 0.95 Å resolution, and the structures of N-terminal methionine and formylmethionine variants of Pf Rd. Contributions of N-terminal interactions to thermostability. *J Biol Inorg Chem* **1998**, *3*, 484–493.

[258] Huang, J.; MacKerell, A. D. CHARMM36 all-atom additive protein force field: Validation based on comparison to NMR data. *J. Comput. Chem.* **2013**, *34*, 2135–2145.

[259] Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781–1802.

[260] Scheurer, M.; Rodenkirch, P.; Siggel, M.; Bernardi, R. C.; Schulten, K.; Tajkhorshid, E.; Rudack, T. PyContact: Rapid, Customizable, and Visual Analysis of Noncovalent Interactions in MD Simulations. *Biophys. J.* **2018**, *114*, 577–583.

[261] Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.* **2011**, *32*, 2319–2327.

[262] Gowers, R. J.; Linke, M.; Barnoud, J.; Reddy, T. J.; Melo, M. N.; Seyler, S. L.; Domański, J.; Dotson, D. L.; Buchoux, S.; Kenney, I. M.; Beckstein, O. MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. *Proc. 15th Python Sci. Conf.* **2016**, 98–105.

[263] Stauch, T.; Dreuw, A. A quantitative quantum-chemical analysis tool for the distribution of mechanical force in molecules. *J. Chem. Phys.* **2014**, *140*, 134107.

[264] Stauch, T.; Dreuw, A. On the use of different coordinate systems in mechanochemical force analyses. *J. Chem. Phys.* **2015**, *143*, 074118.

[265] Stauch, T.; Dreuw, A. Quantum Chemical Strain Analysis For Mechanochemical Processes. *Acc. Chem. Res.* **2017**, *50*, 1041–1048.

[266] Gräter, F. In *Modeling of Molecular Properties*, 1st ed.; Comba, P., Ed.; Wiley-VCH Verlag GmbH & Co. KGaA: Weinheim, 2011; Chapter 19, pp 301–310.

[267] Seifert, C.; Gräter, F. Protein mechanics: How force regulates molecular function. *Biochimica et Biophysica Acta (BBA) - General Subjects* **2013**, *1830*, 4762–4768.

[268] Perdew, J. P. Density-functional approximation for the correlation energy of the inhomogeneous electron gas. *Phys. Rev. B* **1986**, *33*, 8822–8824.

[269] Thom, A. J.; Sundstrom, E. J.; Head-Gordon, M. LOBA: A localized orbital bonding analysis to calculate oxidation states, with application to a model water oxidation catalyst. *Phys. Chem. Chem. Phys.* **2009**, *11*, 11297.

[270] Khaliullin, R. Z.; Cobar, E. A.; Lochan, R. C.; Bell, A. T.; Head-Gordon, M. Unravelling the Origin of Intermolecular Interactions Using Absolutely Localized Molecular Orbitals. *J. Phys. Chem. A* **2007**, *111*, 8753–8765.

[271] Wolinski, K.; Baker, J. Geometry optimization in the presence of external forces: A theoretical model for enforced structural changes in molecules. *Mol. Phys.* **2010**, *108*, 1845–1856.

[272] Stauch, T. Mechanical Switching of Aromaticity and Homoaromaticity in Molecular Optical Force Sensors for Polymers. *Chem. Eur. J.* **2018**, *24*, 7340–7344.

[273] Hermanns, V.; Scheurer, M.; Kersten, N.; Abdellaoui, C.; Wachtveitl, J.; Dreuw, A.; Heckel, A. Re-Thinking Uncaging: A New Antiaromatic Photocage Driven by a Gain of Resonance Energy. *Chem. Eur. J.* **2021**,

[274] Klán, P.; Šolomek, T.; Bochet, C. G.; Blanc, A.; Givens, R.; Rubina, M.; Popik, V.; Kostikov, A.; Wirz, J. Photoremovable Protecting Groups in Chemistry and Biology: Reaction Mechanisms and Efficacy. *Chem. Rev.* **2012**, *113*, 119–191.

[275] Weinstain, R.; Slanina, T.; Kand, D.; Klán, P. Visible-to-NIR-Light Activated Release: From Small Molecules to Nanomaterials. *Chem. Rev.* **2020**, *120*, 13135–13272.

[276] Roxin, A.; Chase, A.; Jeffers, E.; Lukeman, M. Photodecaging from 9-substituted 2,7-dihydroxy and dimethoxyfluorenes: Competition between heterolytic and homolytic pathways. *Photochem. Photobiol. Sci.* **2011**, *10*, 920.

[277] Reinfelds, M.; von Cosel, J.; Falahati, K.; Hamerla, C.; Slanina, T.; Burghardt, I.; Heckel, A. A New Photocage Derived from Fluorene. *Chem. Eur. J.* **2018**, *24*, 13026–13035.

[278] Wan, P.; Krogh, E. Evidence for the generation of aromatic cationic systems in the excited state. Photochemical solvolysis of fluoren-9-ol. *J. Chem. Soc., Chem. Commun.* **1985**, 1207.

[279] Wan, P.; Krogh, E. Contrasting photosolvolytic reactivities of 9-fluorenol vs. 5-suberenol derivatives. Enhanced rate of formation of cyclically conjugated four. pi. carbocations in the excited state. *J. Am. Chem. Soc.* **1989**, *111*, 4887–4895.

[280] Gurzadyan, G. G.; Steenken, S. Solvent-Dependent C−OH Homolysis and Heterolysis in Electronically Excited 9-Fluorenol: The Life and Solvation Time of the 9-Fluorenyl Cation in Water. *Chem. Eur. J.* **2001**, *7*, 1808–1815.

[281] Buck, A. T.; Beck, C. L.; Winter, A. H. Inverted Substrate Preferences for Photochemical Heterolysis Arise from Conical Intersection Control. *J. Am. Chem. Soc.* **2014**, *136*, 8933–8940.

[282] Weigend, F.; Häser, M.; Patzelt, H.; Ahlrichs, R. RI-MP2: Optimized auxiliary basis sets and demonstration of efficiency. *Chem. Phys. Lett.* **1998**, *294*, 143–152.

[283] Shao, Y.; Head-Gordon, M.; Krylov, A. I. The spin–flip approach within time-dependent density functional theory: Theory and applications to diradicals. *J. Chem. Phys.* **2003**, *118*, 4807–4818.

[284] Lange, A. W.; Herbert, J. M. A smooth, nonsingular, and faithful discretization scheme for polarizable continuum models: The switching/Gaussian approach. *J. Chem. Phys.* **2010**, *133*, 244111.

[285] Lange, A. W.; Herbert, J. M. Symmetric versus asymmetric discretization of the integral equations in polarizable continuum solvation models. *Chem. Phys. Lett.* **2011**, *509*, 77–87.

[286] Hodecker, M.; Thielen, S. M.; Liu, J.; Rehn, D. R.; Dreuw, A. Third-Order Unitary Coupled Cluster (UCC3) for Excited Electronic States: Efficient Implementation and Benchmarking. *J. Chem. Theory Comput.* **2020**, *16*, 3654–3663.

[287] Norman, P.; Dreuw, A. Simulating X-ray Spectroscopies and Calculating Core-Excited States of Molecules. *Chem. Rev.* **2018**, *118*, 7208–7248.

[288] Dempwolff, A. L.; Schneider, M.; Hodecker, M.; Dreuw, A. Efficient implementation of the non-Dyson third-order algebraic diagrammatic construction approximation for the electron propagator for closed- and open-shell molecules. *J. Chem. Phys.* **2019**, *150*, 064108.

[289] Dempwolff, A. L.; Belogolova, A. M.; Trofimov, A. B.; Dreuw, A. Intermediate state representation approach to physical properties of molecular electron-attached states: Theory, implementation, and benchmarking. *J. Chem. Phys.* **2021**, *154*, 104117.

Publication List

18. E. Epifanovsky, [...], M. Scheurer, [...], and A. I. Krylov, “Software for the frontiers of quantum chemistry: An overview of developments in the Q-Chem 5 package”, *J. Chem. Phys.* **2021**, In Press, DOI: 10.1063/5.0055522.

17. V. Hermanns, M. Scheurer, N. Kersten, C. Abdellaoui, J. Wachtveitl, A. Dreuw, and A. Heckel, “Re-Thinking Uncaging: A New Antiaromatic Photocage Driven by a Gain of Resonance Energy”, *Chem. Eur. J.* **2021**, Accepted Author Manuscript. DOI: 10.1002/chem.202102351.

16. M. Scheurer, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted, “Efficient Open-Source Implementations of Linear-Scaling Polarizable Embedding: Use Octrees to Save the Trees”, *J. Chem. Theory Comput.* **2021**, *17 (6)*, 3445–3454.

15. D. R. Rehn, Z. Rinkevicius, M. F. Herbst, X. Li, M. Scheurer, M. Brand, A. L. Dempwolff, I. E. Brumboiu, T. Fransson, A. Dreuw, and P. Norman, “Gator: A python-driven program for spectroscopy simulations using correlated wave functions”, *WIREs Comput Mol Sci.* **2021**, *e1528.*

14. M. Scheurer, A. Dreuw, E. Epifanovsky, M. Head-Gordon, and T. Stauch, “Modeling Molecules under Pressure with Gaussian Potentials”, *J. Chem. Theory Comput.* **2021**, *17 (1)*, 583-597.

13. Y. Becker, S. Roth, M. Scheurer, A. Jakob, D. A. Gacek, P. J. Walla, A. Dreuw, J. Wachtveitl, and A. Heckel, “Selective modification for Red-Shifted excitability: A small change in structure, a huge change in photochemistry”, *Chem. Eur. J.* **2020**, *27 (6)*, 2212-2218.

12. M. Scheurer, “Polarizable Embedding for the Algebraic-Diagrammatic Construction Scheme”, *Springer Fachmedien Wiesbaden*, **2020**.

11. M. Scheurer, T. Fransson, P. Norman, A. Dreuw, and D. R. Rehn, “Complex Excited State Polarizabilities in the ADC/ISR Framework”, *J. Chem. Phys.* **2020**, *153*, 074112.

10. M. Scheurer, A. Dreuw, M. Head-Gordon, and T. Stauch, "The Rupture Mechanism of Rubredoxin is More Complex Than Previously Thought", *Chem. Sci.* **2020**, *11*, 6036-6044.

9. D. G. A. Smith, [...], M. Scheurer, [...], and C. D. Sherrill, "Psi4 1.4: Open-source software for high-throughput quantum chemistry", *J. Chem. Phys.* **2020**, *152*, 184108.

8. Q. Sun, [...], M. Scheurer, [...], and G. K.-L. Chan, "Recent developments in the PySCF program package", *J. Chem. Phys.* **2020**, *153*, 024109.

7. M. F. Herbst, M. Scheurer, T. Fransson, D. R. Rehn, and A. Dreuw, "adcc: A Versatile Toolkit for Rapid Development of Algebraic-Diagrammatic Construction Methods", *WIREs Comput Mol Sci.* **2020**, *10:e1462.*

6. Z. Rinkevicius, [...], M. Scheurer, M. Scott, A. Dreuw, and P. Norman, "VeloxChem: A python-driven density-functional theory program for spectroscopy simulations in high-performance computing environments", *WIREs Comput Mol Sci.* **2019**, *10(5):e1457.*

5. M. Scheurer, P. Reinholdt, E. R. Kjellgren, J. M. H. Olsen, A. Dreuw, and J. Kongsted, "CPPE: An Open-Source C++ and Python Library for Polarizable Embedding", *J. Chem. Theory Comput.* **2019**, *15 (11)*, 6154-6163.

4. M. Scheurer, M. F. Herbst, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted, "Polarizable Embedding Combined with the Algebraic Diagrammatic Construction: Tackling Excited States in Biomolecular Systems", *J. Chem. Theory Comput.* **2018**, *14 (9)*, 4870-4883.

3. M. C. R. Melo, R. C. Bernardi, T. Rudack, M. Scheurer, C. Riplinger, J. C. Phillips, J. D. C. Maia, G. B. Rocha, J. V. Ribeiro, J. E. Stone, F. Neese, K. Schulten, and Z. Luthey-Schulten, "NAMD goes quantum: An integrative suite for hybrid simulations", *Nat Methods* **2018**, *15 (5)*, 351-354.

2. M. Scheurer, P. Rodenkirch, M. Siggel, R. C. Bernardi, K. Schulten, E. Tajkhorshid, and T. Rudack, "PyContact: Rapid, customizable, and visual analysis of noncovalent interactions in MD simulations", *Biophys. J.* **2018**, *114 (3)*, 577-583.

1. M. Scheurer, D. Brisker-Klaiman, and A. Dreuw, "Molecular mechanism of flavin photoprotection by archaeal dodecin: Photoinduced electron transfer and Mg^{2+}-promoted proton transfer", *J. Phys. Chem. B* **2017**, *121 (46)*, 10457-10466.

Appendix

Matrix-Vector Product of an ISR One-Particle Operator

Based on the original equations from Trofimov and Schirmer,[40] I derived the matrix-vector product of the ISR one-particle operator with a vector **v** through second order in perturbation theory, i.e., $\mathbf{B}\mathbf{v} = \mathbf{r}$. The **B** matrix possesses a block structure similar to that of the ADC matrix,[40] such that the vectors **v** and **r** contain a singles and a doubles block. Within the equations, anti-symmetrized two-electron integrals $\langle pq||rs\rangle$ occur, together with the T2-amplitudes defined as

$$t_{ij}^{ab} = \frac{\langle ij||ab\rangle}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j}.$$

Furthermore, the MP2 density matrix contribution[34] is defined as

$$\rho_{ia}^{(2)} = -\frac{1}{2\left(\varepsilon_a - \varepsilon_i\right)} \left[\sum_{jbc} t_{ij}^{bc} \langle ja||bc\rangle + \sum_{jkb} t_{jk}^{ab} \langle jk||ib\rangle \right].$$

Parts of this chapter have already been published in:

- M. Scheurer, T. Fransson, P. Norman, A. Dreuw, and D. R. Rehn, "Complex Excited State Polarizabilities in the ADC/ISR Framework", *J. Chem. Phys.* **2020**, *153*, 074112. (Reference 41)

The permutation operator $\hat{\mathcal{P}}_{pr,qs}$ permutes the index pairs (p, q) and (r, s). Collecting all terms, one finds the result for the singles block of the matrix-vector product as

$$\begin{aligned}
r_{ai} =& \sum_c d_{ac} v_{ci} - \sum_k d_{ik} v_{ak} - \sum_c v_{ci} \left(\sum_j \rho^{(2)}_{ja} d_{cj} + \rho^{(2)}_{jc} d_{aj} \right) \\
& - \sum_k v_{ak} \left(\sum_b \rho^{(2)}_{ib} d_{bk} + \rho^{(2)}_{kb} d_{bi} \right) \\
& - \frac{1}{4} \sum_c v_{ci} \left(\sum_{efmn} t^{ef}_{mn} \left(t^{af}_{mn} d_{ec} + t^{cf}_{mn} d_{ea} \right) \right) \\
& + \sum_c v_{ci} \left(-\frac{1}{2} \sum_{efmn} t^{ce}_{mn} t^{af}_{mn} d_{ef} + \sum_{fmnj} t^{cf}_{mn} t^{af}_{jn} d_{jm} \right) \\
& + \frac{1}{4} \sum_k v_{ak} \left(\sum_{efmn} t^{ef}_{mn} \left(t^{ef}_{in} d_{km} + t^{ef}_{kn} d_{im} \right) \right) \\
& + \sum_k v_{ak} \left(- \sum_{edfn} t^{ef}_{kn} t^{df}_{in} d_{ed} + \frac{1}{2} \sum_{efmn} t^{ef}_{kn} t^{ef}_{im} d_{mn} \right) \\
& + \frac{1}{2} \sum_{ck} v_{ck} \left(\left(1 + \hat{\mathcal{P}}_{ai,ck} \right) \left(\sum_{efn} t^{ef}_{kn} t^{af}_{in} d_{ec} - \sum_{fmn} t^{cf}_{mn} t^{af}_{in} d_{km} \right) \right) \\
& + \sum_{ck} v_{ck} \left(- \sum_{fmn} t^{cf}_{kn} t^{af}_{im} d_{mn} + \sum_{efn} t^{ce}_{kn} t^{af}_{in} d_{ef} \right) \\
& - 2 \sum_{dl} v_{adil} \left(d_{ld} - \sum_{fn} t^{df}_{ln} d_{fn} \right) + 2 \sum_{cl} v_{cail} \left(d_{lc} - \sum_{fn} t^{cf}_{ln} d_{fn} \right) \\
& - 2 \sum_{dkl} v_{adkl} \sum_e t^{ed}_{kl} d_{ei} - 2 \sum_{cdl} v_{cdil} \sum_n t^{cd}_{nl} d_{an} .
\end{aligned}$$

The doubles part of the vector is given by

$$\begin{aligned}
r_{abij} = &- v_{ai} \left(d_{jb} - \sum_{fn} t_{jn}^{bf} d_{fn} \right) + v_{aj} \left(d_{ib} - \sum_{fn} t_{in}^{bf} d_{fn} \right) \\
&+ v_{bi} \left(d_{ja} - \sum_{fn} t_{jn}^{af} d_{fn} \right) - v_{bj} \left(d_{ia} - \sum_{fn} t_{in}^{af} d_{fn} \right) \\
&- \sum_{k} v_{ak} \sum_{e} t_{ij}^{eb} d_{ek} + \sum_{k} v_{bk} \sum_{e} t_{ij}^{ea} d_{ek} - \sum_{c} v_{ci} \sum_{n} t_{nj}^{ab} d_{cn} + \sum_{c} v_{cj} \sum_{n} t_{ni}^{ab} d_{cn} \\
&+ 2 \sum_{c} d_{ac} v_{cbij} - d_{bc} v_{caij} - 2 \sum_{k} d_{ki} v_{abkj} - d_{kj} v_{abki}.
\end{aligned}$$

Danksagung

An erster Stelle möchte ich mich bei Prof. Dr. Andreas Dreuw bedanken, der mir die Möglichkeit gegeben hat, diese Dissertation in seiner Arbeitsgruppe anzufertigen. Durch sein Vertrauen und den Freiraum, den er mir bei meiner Arbeit geboten hat, konnte ich in den letzten drei Jahren spannende Themenstellungen bearbeiten und meine Fähigkeiten und Interessen vielfältig entwickeln. Ich habe mich von Anfang an in der Gruppe wohlgefühlt und dank Andreas ist es mir leicht gefallen, in meinem Forschungsgebiet Fuß zu fassen. Sein Input hat mich dabei nicht nur wissenschaftlich, sondern auch persönlich weitergebracht.

I am grateful to Prof. Dr. Jacob Kongsted who welcomed me in his research group during my Master thesis and initially spurred my interest in the development of polarizable models. Without the continued collaborations and his support, a major part of my work would not have been possible.

I thank Peter Reinholdt for very exciting and fruitful collaboration during the last years. It has always been a great pleasure for me to jointly work on PE-related application and development projects.

Mein Dank gilt außerdem Prof. Dr. Oriol Vendrell für die Begutachtung dieser Arbeit. Zudem möchte ich Dr. Michael F. Herbst für unsere Zusammenarbeit am `adcc`-Projekt danken, die mir stets viel Freude bereitet hat. Durch hilfreiche Diskussionen und etliche PR-Reviews auf GitHub konnte ich sehr viel Neues über Python und Softwaredesign lernen und schlussendlich meine entwickelten Methoden mit `adcc` implementieren. Mein besonderer Dank gilt Dr. Dirk R. Rehn für gemeinsame Response-Projekte und die zugehörigen Post-It-Herleitungen. Außerdem bin ich Prof. Dr. Tim Stauch für zwei sehr spannende und anspruchsvolle Kollaborationsprojekte dankbar, die zu dieser Dissertation beigetragen haben. Über diese Projekte hinaus hat Tim immer ein offenes Ohr für meine Fragen gehabt und mir auch bei meiner eigenen Forschung durch seinen Input und Diskussionen oft geholfen.

Der DFG-Graduiertenschule „Complex Light Control“ (CLiC) danke ich herzlich für die finanzielle Unterstützung. Meinen CLiC-Kollaborationspartnern – Volker Hermanns, Chahinez Abdellaoui, Yvonne Becker und Sina Roth – danke ich für die gelungene Zu-

sammenarbeit, die für mich sehr abwechslungsreich und lehrreich war. Zudem bin ich Prof. Dr. Alexander Heckel und Prof. Dr. Josef Wachtveitl als den Mitgliedern meines CLiC-Thesis-Advisory-Komitees für ihr Mentoring zum Dank verpflichtet.

Für das Korrekturlesen dieser Dissertation bedanke ich mich bei Adrian Dempwolff, Marvin Hoffmann, Dirk R. Rehn, Antonia Papapostolou, Tim Stauch, Manuel Hodecker und Linnéa Rohde. Der gesamten Arbeitsgruppe danke ich für eine gute Arbeitsatmosphäre. Insbesondere die Diskussionen mit Adrian, Manuel, Marvin, Sebastian und Dirk haben zum Gelingen dieser Dissertation beigetragen. Nicht zuletzt danke ich allen Studierenden, die ich während der letzten drei Jahre betreuen durfte, im Besonderen Anna-Lena Fertig, Anna Weidlich, Antonia Papapostolou und Jonas Leitner, für ihre Motivation und Impulse, die für meine Forschung sehr hilfreich waren. Ebenso gebührt mein Dank Ellen Vogel für ihre verlässliche Hilfe bei organisatorischen und verwaltungstechnischen Problemen sowie Manfred Trunk für die hervorragende IT-Administration.

Meinen Freunden Nicolas und Lucas danke ich für viele entspannte „Feierabendrunden“ auf dem Rennrad, Simeon für kulinarische Highlights und Adrian für gemeinsame Werkstattabende.

Linnéa bin ich zutiefst dankbar für ihre bedingungslose Unterstützung und Hilfe, die mir in schwierigen Situationen und vor allem beim Schreiben dieser Dissertation viel Kraft, Ausdauer und die nötige Ruhe gegeben haben.

Zum Schluss möchte ich mich ganz herzlich bei meinen Eltern bedanken, denen diese Dissertation, zusammen mit meinem Bruder Lukas, gewidmet ist. Ohne deren Unterstützung während meines Studiums und der Promotion in jeglicher Hinsicht wäre diese Arbeit nicht möglich gewesen.

Eidesstattliche Versicherung gemäß § 8 der Promotionsordnung für die Naturwissenschaftlich-Mathematische Gesamtfakultät der Universität Heidelberg

1. Bei der eingereichten Dissertation zu dem Thema
   Quantum Chemical Methods for Molecular Properties in Complex Environments
   handelt es sich um meine eigenständig erbrachte Leistung.
2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.
3. Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.
4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.
5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt.

Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Ort und Datum	Unterschrift