DISS ETH NO. 22363

Developments for a Relativistic Many-1/2-Fermion Theory

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

BENJAMIN SIMMEN

MSc Chemistry, ETH Zürich

born on 27.06.1986

citizen of Erlach BE and Italy

accepted on the recommendation of

Prof. Dr. Markus Reiher
Prof. Dr. Frédéric Merkt

2014

to my beautiful wife

Contents

Contents

Zusammenfassung

Genaue Studien für Systeme bestehend aus subatomaren Teilchen sind von fundamentalem Interesse für viele Bereiche der Physik und der Chemie. Die grundlegenste Theorie für Systeme von elektrisch geladenen Teilchen ist die Quantenelektrodynamik. Sie quantisiert das Materie-Feld und das Strahlungs-Feld und verhält sich konform mit der speziellen Relativitätstheorie. Jedoch sind Berechnungen basierend auf der Quantenelektrodynamik aufwändig. Für Systeme, in welchen Strahlung auch klassisch beschrieben werden kann, ist es möglich, die Quantisierung des Strahlungs-Feldes zu vernachlässigen.

Der Dirac Hamiltonian liefert eine Beschreibung für Systeme von Fermionen mit Spin 1/2, in denen nur das Materie-Feld quantisiert ist, aber wichtige Aspekte der speziellen Relativitätstheorie wie die Geschwindigkeitsabhängigkeit der Masse und magnetische und Retardations-Effekte korrekt beschrieben werden. Das Hauptproblem des Dirac Hamiltonians ist das nach unten unbeschränkte Energiespektrum. Eine Lösung für dieses Problem ist die sogenannte ”Kinetic-Balance”. Leider ist nur eine Orbital-basierte Form bekannt, jedoch keine explizit korrelierte. Ein Teil dieser Arbeit widmet sich der Herleitung einer explizit korrelierten ”Kinetic-Balance”.

Der Dirac Hamiltonian ist die Grundlage der relativistischen Elektronenstrukturtheorie. Hier nutzt man die Born–Oppenheimer-Näherung. Translationseffekte müssen in diesem Fall nicht berücksichtigt werden. In der nicht-relativistischen pre-Born–Oppenheimer-Theorie können Translationseffekte mit Hilfe einer linearen Transformation der Kartesischen Koordinaten eliminiert werden indem man die Kartesischen Koordinaten des Massenschwerpunkts separiert. Eine solche lineare Transformation existiert nicht für die relativistische Theorie. Als Alternative präsentieren wir translationsinvariante Integralausdrücke, welche gleichermassen auf relativistische wie nicht-relativistische Probleme angewendet werden können.

Am Ende dieser Arbeit kombinieren wir die explizit korrelierte ”Kinetic-Balance” mit den translationsinvarianten Integralausdrücken. Dies liefert eine erst-quantisierte relativistische viel-1/2-Fermionen Theorie frei von Translationskontamination falls notwendig. Wir verwenden diese Theorie um das Feinstrukturenergiespektrum von Wasserstoff und wasserstoffähnlichen Ionen zu studieren. Unsere Resultate sind numerisch exakt.

Abstract

The accurate study of systems composed of subatomic particles is of fundamental interest to many branches of physics and chemistry. The most fundamental theory for systems of electrically charged subatomic particles is quantum electrodynamics. It quantizes both the matter-field and the radiation field and is fully compliant with special relativity. Yet, calculations based on quantum electrodynamics are cumbersome. For systems where the radiation can be described classically, the quantization of the radiation field can be neglected.

The Dirac Hamiltonian provides a covariant description of systems of fermions with spin 1/2 where only the matter field is quantized but fundamental aspects of special relativity such as mass-velocity effects and magnetic and retardation effects are maintained. The main issue with the Dirac Hamiltonian is the unboundedness of the energy spectrum. A solution to this problem is the kinetic-balance condition. However, only an orbital-based form is known but no explicitly correlated form. One part of this work is dedicated to the derivation of an explicitly correlated kinetic-balance condition.

The Dirac Hamiltonian is the foundation of relativistic electronic structure theory. Here, the Born–Oppenheimer approximation is employed. Translational effects do not need to be taken into account. In the non-relativistic pre-BO theory, it is possible to perform a linear transformation of the Cartesian coordinates to separate the center-of-mass Cartesian coordinate to eliminate translational effects. No such transformation exists for the relativistic theory. As an alternative, we present a scheme of translationally invariant integrals which can be equally employed to the relativistic and non-relativistic framework as it only considers translational effects but does not explicitly rely on the center-of-mass coordinate.

Finally, we combine the explicitly correlated kinetic-balance with the framework of translationally invariant integrals. This results in a relativistic many-1/2-fermion theory free from translational contaminations if desired. We use this framework to study the fine-structure spectrum of atomic hydrogen and hydrogen-like ions. Our results are numerically exact.

1

Introduction

Explicitly correlated trial wave functions contain factors which depend on the pairwise inter-particle distances. These factors improve the quality of the approximation of the state function significantly. In electronic structure theory, explicitly correlated basis functions are used in F12 and R12 methods to approach the basis set limit without having to rely on very large basis sets [1].

The non-relativistic description of few-body systems in terms of quantum mechanics is based on the Schrödinger Hamiltonian. Methods based on basis-set expansions for approximating the many-body state function have been developed during the last few decades [2, 3]. Here, the Born–Oppenheimer (BO) approximation [4–6] is not invoked and the parametrization of the trial wave function treats all particles on the same footing. Such methods are known as non-Born–Oppenheimer methods. Here, we use pre-Born–Oppenheimer (pre-BO) methods in order to stress that the BO approximation is not invoked. Due to the correlation between the particles, explicitly correlated basis functions are of key importance for the generation of accurate trial wave functions.

Quantum electrodynamics (QED) is the most fundamental theory for systems of electrically charged particles. It is fully Lorentz-covariant and therefore complies with special relativity. Yet, calculations based on QED are cumbersome. However, for chemical systems, only the quantization of the matter field is important and particle pair-creation and -annihilation effects are not relevant. Incorporating QED effects by perturbation theory is a common method and provides a well-understood framework for the relativistic description. Yet,

there are certain drawbacks. For instance, perturbative methods do not exhibit variational stability. Also, it is not known if high-order corrections are accurately included or if the method breaks down.

A different approach is based on the Dirac Hamiltonian which provides a first-quantized description of relativity suited for the description of chemical systems [7–13]. It is the cornerstone of relativistic quantum chemistry and, while not being fully Lorentz-covariant, it captures relativistic effects up to second order in terms of the fine-structure constant α for the Dirac-Coulomb Hamiltonian. Here, all interactions are considered instantaneous in terms of the Coulomb interaction. Magnetic and retardation effects due to the finite speed of light can be considered by the Breit interaction. Then, relativistic effects are correctly included up to fourth order in the fine structure constant.

The unboundedness of the Dirac Hamiltonian requires certain precautions in order to ensure variational stability of calculations. For trial wave functions approximated as products of orbitals, variational stability can be ensured by generating the model spaces in terms of the kinetic-balance condition [14–25]. The first problem for the development of a relativistic many-fermion theory is the formulation of an explicitly correlated variant of the kinetic-balance condition.

Once variational stability is ensured, the problem of translational invariance has to be addressed. Each observable, such as the total energy or transition dipole moments, can be separated into a translationally invariant and a translationally dependent part. Only the translationally invariant part is of interest since it describes the internal properties of the molecule. Thus, the translationally dependent part has to be eliminated. In the non-relativistic theory, the translational behavior of the center-of-mass Cartesian coordinate (CMCC) describes the translation of the total system. Then, a linear combination of the laboratory-fixed Cartesian coordinates (LFCC) which separates the CMCC from some set of translationally invariant Cartesian coordinates (TICC) [26] can be used to formulate a translationally invariant Hamiltonian [27]. The CMCC does, however, not relate to the translation of a relativistic system so that such a separation does not yield the desired effect [28]. A related transformation involves the center-of-momentum frame [28], where the total momentum of the system is zero. This is fairly straightforward but leads to the problem that the non-relativistic and the relativistic framework rely on different schemes for the elimination of the translational contamination. Therefore, the issue of translational invariance has to be addressed anew in order to formulate a framework which suits the relativistic and non-relativistic theory equally.

In this thesis, we address the two main issues regarding a relativistic first-quantized many-1/2-fermion theory which obeys the variational principle and is free from translational contaminations.

2

Non-Relativistic pre-Born–Oppenheimer Theory

In this chapter, we focus on the non-relativistic pre-BO framework on which we will later ground our relativistic framework [29, 30]. For a detailed presentation, we refer the reader to Refs. [3, 31, 32], and the book by Suzuki and Varga [2].

2.1 Schrödinger Hamiltonian

The N-particle Schrödinger Hamiltonian

$$\hat{H} = \hat{T} + \hat{V} \tag{2.1}$$

is the basis of the non-relativistic pre-BO framework. We generally rely on Hartree atomic units unless otherwise stated. It consists of the kinetic-energy operator

$$\hat{T} = -\sum_{i=1}^{N} \frac{1}{2m_i} \Delta_{\boldsymbol{r}_i} \ , \tag{2.2}$$

where m_i is the mass of particle i and $\Delta_{\boldsymbol{r}_i}$ is the Laplacian with respect to $\boldsymbol{r}_i$, the Cartesian coordinates of particle i , and the potential-energy operator $\hat{V}$. The potential energy consists of the Coulomb interaction among pairs of particles

$$\hat{V}_C = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{q_i q_j}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} \tag{2.3}$$

where q_i and q_j are the charges of the interacting particles, and any external fields present. The state function $\boldsymbol{\Psi}(\boldsymbol{r})$, where $\boldsymbol{r}^{\mathrm{T}} = (\boldsymbol{r}_1^{\mathrm{T}}, \ldots, \boldsymbol{r}_N^{\mathrm{T}})$ collects all one-particle Cartesian coordinates, is determined together with the energy through the Rayleigh factor

$$E[\boldsymbol{\Psi}] = \frac{\langle \boldsymbol{\Psi} | \, \hat{H} \, | \boldsymbol{\Psi} \rangle}{\langle \boldsymbol{\Psi} \, | \boldsymbol{\Psi} \rangle} \ . \tag{2.4}$$

The state functions are eigenfunctions of the Hamiltonian in Eq. (2.1) and are therefore critical points of the Rayleigh factor. Additionally, the variational principle states that any approximation of the ground-state function has an energy which is higher than the exact ground-state energy. Thus, minimization of the energy is a valid strategy of optimizing any approximation to the state function.

The energies of the Hamiltonian in Eq. (2.1) contains a contamination from translational effects because the Hamiltonian is not translationally invariant. A method of eliminating these translational effects is to perform a linear transformation $\boldsymbol{U}_x$ of the original $\boldsymbol{r}$ to a set of TICC $\boldsymbol{x}$ and the CMCC $\boldsymbol{x}_{\mathrm{CM}}$ [26]

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_{\mathrm{CM}} \end{bmatrix} = \boldsymbol{U}_x \boldsymbol{r} \quad \text{and} \quad \boldsymbol{r} = \boldsymbol{U}_x^{-1} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_{\mathrm{CM}} \end{bmatrix} \ . \tag{2.5}$$

The matrix $\boldsymbol{U}_x$ is non-singular with the general super structure

$$\boldsymbol{U}_x = \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{\mu} \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{U}_x^{-1} = \begin{bmatrix} \boldsymbol{M}' & \boldsymbol{\epsilon} \end{bmatrix} \tag{2.6}$$

where the parts $\boldsymbol{M}$, a $(N-1) \times N$ matrix, and $\boldsymbol{M}'$, a $N \times (N-1)$ matrix, depend on the choice of translationally invariant coordinates, $\boldsymbol{\mu}_i = m_i / m_{tot}$ is a vector of dimension N containing all mass fractions and $\boldsymbol{\epsilon}_i = 1$ is a vector of dimension N with 1 as each entry. $\boldsymbol{M}$ and $\boldsymbol{M}'$ depend on the choice of translationally invariant coordinates and obey the restrictions

$$\sum_{j=1}^{N} (\boldsymbol{U}_x)_{ij} = 0, \quad \text{with} \quad i = 1, 2, \ldots, N-1 \tag{2.7}$$

and

$$(\boldsymbol{U}_x)_{N,j} = m_j / m_{\mathrm{tot}}, \quad \text{with} \quad j = 1, 2, \ldots, N \ . \tag{2.8}$$

This guarantees the translational invariance for the coordinates $\boldsymbol{x}$.

Transforming the Hamiltonian in Eq. (2.1) to some set of TICC leads to the translationally invariant Hamiltonian and the center-of-mass kinetic-energy operator

$$\hat{H}_{\mathrm{TICM}} = \sum_{i=1}^{N-1}\sum_{j=1}^{N-1} \boldsymbol{\mathcal{M}}_{ij} \nabla_{\boldsymbol{x}_i}^{\mathrm{T}} \nabla_{\boldsymbol{x}_i} + \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{q_i q_j}{\left(\boldsymbol{f}_{ij} \otimes \boldsymbol{I}_3\right)^{\mathrm{T}} \boldsymbol{x}} + \hat{T}_{\mathrm{CM}} \tag{2.9}$$

where

$$\boldsymbol{\mathcal{M}}_{ij} = \sum_{k=1}^{N} \left(\boldsymbol{U}_x^{-1}\right)_{ik} \left(\boldsymbol{U}_x^{-1}\right)_{jk} / (2m_k) \tag{2.10}$$

and

$$(\boldsymbol{f}_{ij})_k = \left(\boldsymbol{U}_x^{-1}\right)_{ik} - \left(\boldsymbol{U}_x^{-1}\right)_{jk} \ . \tag{2.11}$$

By eliminating the center-of-mass kinetic-energy operator

$$\hat{T}_{\mathrm{CM}} = -\frac{1}{2m_{\mathrm{tot}}} \Delta_{\boldsymbol{x}_{\mathrm{CM}}} \tag{2.12}$$

from the transformed Hamiltonian in Eq. (2.9) we obtain the translationally invariant (TI) Hamiltonian as

$$\hat{H}_{\mathrm{TI}} = \hat{H}_{\mathrm{TICM}} - \hat{T}_{\mathrm{CM}} \ . \tag{2.13}$$

As an alternative, this Hamiltonian can also be obtained from the condition that the combined momenta of all particles is zero [33].

The translationally invariant energy is accordingly obtained by linearly transforming the Rayleigh factor in Eq. (2.4) by transforming both the Hamiltonian and the state function and then eliminating the center-of-mass Cartesian coordinate $\boldsymbol{x}_{\mathrm{CM}}$. This is the basis of the traditional pre-BO theory [3, 30, 34].

2.2 Approximations of the State Function

The state function of the LFCC Hamiltonian in Eq. (2.1) can be separated in terms of a spin part and a spatial part. The spatial part can then be separated into an angular part and a radial part. The spin part can be formed from the elementary spin states of each particle type, as the spin of the particles only interact among the same type of particles, e.g. only electron spins interact with each other but they do not interact with the proton spins. The elementary spin function $\boldsymbol{\chi}_t^{S_t, M_{S_t}}$ for N_t particles of type t can be calculated recursively

$$\boldsymbol{\chi}_t^{S_t, M_{S_t}}(N_t) = \sum_{i=-S_1}^{S_1} c(X, i | S_t, M_{S_t}, S_1) \boldsymbol{\chi}_t^{S_t - X, M_{St} - i}(N_t - 1) \otimes \boldsymbol{\chi}_t^{S_1, i}(1) \ , \tag{2.14}$$

where X is to be chosen such that $S_t - X < (N_t - 1)S_t$, through angular momentum recoupling using the Clebsch–Gordan coefficients

$$c(X, i|S_t, M_{S_t}, S_1) = \langle S_t - X, M_{S_t} - i; S_1, i|S_t, M_{S_t}\rangle \tag{2.15}$$

as expansion coefficients. S_1 is the spin of a single particle, S_t and M_{S_t} are the spin quantum numbers of the particle ensembles. The ending condition of the recursion relation are the one-particle spin states

$$\left(\boldsymbol{\chi}_t^{S_1,M_{S_1}}(1)\right)_i = \delta_{i,M_{S_1}+S_1+1} \ . \tag{2.16}$$

The total spin state for m ensembles of particles is formed as the direct product the elementary spin functions

$$\boldsymbol{\chi}^{\{S_t\},\{M_{S_t}\}}(\{N_t\}) = \boldsymbol{\chi}_1^{S_1,M_{S_1}}(N_1) \otimes \ldots \otimes \boldsymbol{\chi}_m^{S_m,M_{S_m}}(N_m) \ . \tag{2.17}$$

Since the spin eigenstates can be represented by vectors, an implementation of Eq. (2.14) only involves standard routines from linear algebra.

Before, we continue with the angular part of the state function, we will shortly discuss the spin operators related to the spin eigenstates determined by Eq. (2.14). For a single particle with spin S, the operator which can be most easily determined is the projection onto the z-axis $\hat{\boldsymbol{s}}_z$. It is a diagonal square matrix with dimension $2S + 1$. The entries are then ranging from $+S$ to $-S$ in steps of 1:

$$(\hat{\boldsymbol{s}}_z)_{ij} = \delta_{ij} \times (S - i) \ . \tag{2.18}$$

The other two projection matrices are more difficult to calculate. It is convenient to introduce the ladder operators $\hat{\boldsymbol{s}}_+$ and $\hat{\boldsymbol{s}}_-$ which are related to the x and y projection as

$$\hat{\boldsymbol{s}}_x = \frac{1}{2}\left(\hat{\boldsymbol{s}}_+ + \hat{\boldsymbol{s}}_-\right) \tag{2.19}$$

$$\hat{\boldsymbol{s}}_y = \frac{i}{2}\left(\hat{\boldsymbol{s}}_+ - \hat{\boldsymbol{s}}_-\right) \ . \tag{2.20}$$

The ladder operators raise or lower the M_S quantum number of the one particle state by ± 1. The matrix representation of $\hat{\boldsymbol{s}}_+$ and $\hat{\boldsymbol{s}}_-$ have dimension $2S + 1$ and the elements are defined as

$$(\hat{\boldsymbol{s}}_+)_{ij} = \delta_{i+1,j} \quad \text{and} \quad (\hat{\boldsymbol{s}}_-)_{ij} = \delta_{i-1,j} \tag{2.21}$$

and all elements of the matrix are zero except the elements directly below ($\hat{\boldsymbol{s}}_+$) or above ($\hat{\boldsymbol{s}}_-$) the diagonal which are 1. The last operator $\hat{\boldsymbol{s}}^2$ can simply be formed as

$$\hat{\boldsymbol{s}}^2 = \hat{\boldsymbol{s}}_x^2 + \hat{\boldsymbol{s}}_y^2 + \hat{\boldsymbol{s}}_z^2 \tag{2.22}$$

from the matrices defined in Eqs. (2.18) – (2.20). For ensembles of particles of the same type, the spin operators can be formed as the direct sum of the one-particle spin operators.

The Hamiltonian in Eq. (2.1) commutes with the total spatial angular momentum operator

$$\hat{L} = \sum_{i=1}^{N} \hat{l}_i \tag{2.23}$$

where $\hat{l}_i$ is the one-particle spatial angular momentum operator of particle i. Therefore, the state function is an eigenfunction of $\hat{L}$. One method of generating the eigenfunction is reminiscent to Eq. (2.14) where $\hat{L}$ is formed recursively through angular momentum recoupling as products of the eigenfunctions of the one-particle spatial angular momentum operators $\hat{l}_i$. But these eigenfunctions are solid harmonics and the evaluation of the integrals will be involved. Suzuki and Varga [2,35,36] have presented a general form for the eigenfunction of $\hat{L}$ for N particles. This form is known as the global vector representation (GVR)

$$\mathcal{Y}_L^{M_L}(\boldsymbol{v}, K) = |\boldsymbol{v}|^{2K+L} Y_L^{M_L}(\hat{\boldsymbol{v}}) \ , \tag{2.24}$$

with

$$\boldsymbol{v} = \boldsymbol{u} \cdot \boldsymbol{r} \ , \tag{2.25}$$

where $\boldsymbol{u}$ is the global-vector and K some non-negative integer.

The radial part $G(\boldsymbol{r})$ of the state function cannot be formulated exactly like the spin part and the spatial angular part and therefore has to be approximated. A reliable approximation of the radial part is a linear combination of square integrable N-particle functions

$$G(\boldsymbol{r}) = \sum_{i=1}^{n} c_i G_i(\boldsymbol{r}) \ . \tag{2.26}$$

A function well suited for representing the radial part are Slater-type functions. However, integral expressions for such functions are difficult to evaluate. Explicitly correlated Gaussian functions (ECGs)

$$G_i(\boldsymbol{r}) = \exp\left(-\frac{1}{2}\boldsymbol{r}^{\mathrm{T}} \left(\boldsymbol{A}_i \otimes \mathbf{1}_3\right) \boldsymbol{r}\right) \tag{2.27}$$

provide us with an alternative to Slater-type functions and lead to integral expressions which are more easily evaluated. $\boldsymbol{A}_i$ is a $N \times N$ positive definite matrix in order to ensure that the metric of the state function is non-vanishing.

Last, we have to enforce particle-exchange symmetry onto the state function. This can be done with the (anti-)symmetrization operator

$$\hat{\mathcal{A}} = \prod_{i}^{n_t} \hat{\mathcal{A}}_i \tag{2.28}$$

where

$$\hat{\mathcal{A}}_i = \sum_{p \in P} \epsilon(p) \hat{P}_p \quad \text{where} \quad \epsilon(p) = \begin{cases} 1 & \text{for bosons} \\ (-1)^p & \text{for fermions} \end{cases} \tag{2.29}$$

is the (anti-)symmetrization operator for each ensemble of a certain type of particles.

Combining the different parts we obtain the approximate state function

$$\begin{aligned} \boldsymbol{\Psi}(\boldsymbol{r}) &= \sum_{i=1}^{N} c_i \boldsymbol{\Phi}^{L}_{M_L\ i}(\boldsymbol{r}; \boldsymbol{A}_i, \boldsymbol{u}_i, K_i) \\ &= \sum_{i=1}^{N} c_i \hat{\mathcal{A}} \boldsymbol{\chi}^{\{S\},\{M_S\}}(\{N_t\}) \mathcal{Y}^{M_L}_{L}(\boldsymbol{v}_i, K_i) G_i(\boldsymbol{r}) \end{aligned} \tag{2.30}$$

where c_i, $\boldsymbol{A}_i$, K_i and $\boldsymbol{u}_i$ are variational parameters for each basis function $\boldsymbol{\Phi}^{L}_{M_L\ i}(\boldsymbol{r}; \boldsymbol{A}_i, \boldsymbol{u}_i, K_i)$ which we are left to optimize such that the energy defined in Eq. (2.4) is minimal in accordance with the variational principle. The optimization of the expansion coefficients c_i can be done analytically since they are linear parameters. Minimization of the energy with respect to the expansion coefficients leads to the generalized eigenproblem:

$$\boldsymbol{HC} = \boldsymbol{ESC} \tag{2.31}$$

where $\boldsymbol{H}$ and $\boldsymbol{S}$ are the matrix representations of the Hamiltonian and the overlap with the elements defined as

$$\boldsymbol{H}_{IJ} = \left\langle \boldsymbol{\Phi}^{L}_{M_L\ I}(\boldsymbol{r}; \boldsymbol{A}_I, \boldsymbol{u}_I, K_I) \right| \hat{H} \left| \boldsymbol{\Phi}^{L}_{M_L\ J}(\boldsymbol{r}; \boldsymbol{A}_J, \boldsymbol{u}_J, K_J) \right\rangle \tag{2.32}$$

$$\boldsymbol{S}_{IJ} = \left\langle \boldsymbol{\Phi}^{L}_{M_L\ I}(\boldsymbol{r}; \boldsymbol{A}_I, \boldsymbol{u}_I, K_I) \right| \left| \boldsymbol{\Phi}^{L}_{M_L\ J}(\boldsymbol{r}; \boldsymbol{A}_J, \boldsymbol{u}_J, K_J) \right\rangle \ . \tag{2.33}$$

The matrix $\boldsymbol{C}$ contains the expansion coefficients for the different vibrational states and the diagonal matrix $\boldsymbol{E}$ contains the corresponding energies. Integral expressions for the non-relativistic kinetic energy can be found in Section 4.1 and expressions for the Coulomb interaction and the overlap are presented in Section 8.2.

The spatial terms of the basis functions described in Eq. (2.30) can also be formulated in terms of a generating function [2]:

$$\boldsymbol{\Phi}_{M_L}^{L}(\boldsymbol{r};\boldsymbol{A},\boldsymbol{u},K)=\frac{1}{B_{KL}}\int d\hat{\boldsymbol{\varepsilon}}Y_{M_L}^{L}(\hat{\boldsymbol{\varepsilon}})\left\{\frac{\partial^{\lambda}}{\partial a^{\lambda}}g(\boldsymbol{r};\boldsymbol{A},a\boldsymbol{u}\otimes\boldsymbol{\varepsilon})\right\}_{a=0,|\boldsymbol{\varepsilon}|=1} \tag{2.34}$$

where $\lambda_i = 2K + L$ and with the generating function

$$g(\boldsymbol{r};\boldsymbol{A},a\boldsymbol{u}\otimes\boldsymbol{\varepsilon})=\exp\left(-\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{A}\otimes\mathbf{1}_3)\boldsymbol{r}+(a\boldsymbol{u}\otimes\boldsymbol{\varepsilon})^{\mathrm{T}}\boldsymbol{r}\right) \tag{2.35}$$

and

$$B_{mL}=\frac{4\pi(L+2m)!(L+m+1)!2^{L+1}}{m!(2L+2m+2)!}\ . \tag{2.36}$$

Note that we have omitted the (anti-)symmetrization operator, the spin part and the basis function index for the sake of brevity. This form is particularly convenient for the derivation of integral expressions.

The approximation of the state function in Eq. (2.30) has the convenient property that its form is conserved under a linear transformation of $\boldsymbol{r}$. Thus, the state function can be transformed to some set of TICC by transformation of $\boldsymbol{A}_i$ and $\boldsymbol{u}_i$ as

$$\boldsymbol{A}^{(x)}=\boldsymbol{U}_x^{-\mathrm{T}}\boldsymbol{A}\boldsymbol{U}_x^{-1}\quad\Leftrightarrow\quad\boldsymbol{A}=\boldsymbol{U}_x^{\mathrm{T}}\boldsymbol{A}^{(x)}\boldsymbol{U}_x \tag{2.37}$$

and

$$\boldsymbol{u}^{(x)}=\boldsymbol{U}_x^{-\mathrm{T}}\boldsymbol{u}\quad\Leftrightarrow\quad\boldsymbol{u}=\boldsymbol{U}_x^{\mathrm{T}}\boldsymbol{u}^{(x)}\ . \tag{2.38}$$

2.2.1 Parametrization Schemes

Depending on the framework, different parametrizations exist. A simple way of parametrizing $\boldsymbol{A}$ and $\boldsymbol{A}^{(x)}$ is related to the Cholesky decomposition

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} \tag{2.39}$$

where $\boldsymbol{L}$ is a lower triangular matrix with the only restriction that its diagonal elements are positive definite. Additionally, based on the work by Boys [37], it is possible to parametrize $\boldsymbol{A}$ as:

$$\boldsymbol{A}_{ij}=\left(\alpha_{ii}+\sum_{l=1}^{N}\beta_{il}\right)\delta_{ij}-\beta_{ij}(1-\delta_{ij}) \tag{2.40}$$

Here the α parameters are positive definite and both the α_{ij} and β_{ij} factors are the new variational parameters.

Furthermore, there is a parametrization originally introduced by Mátyus [30] which will later become important to ensure translational invariance:

$$(\boldsymbol{A})_{ij} = -\alpha_{ij}(1-\delta_{ij}) + \left(\sum_{k=1,k\neq i}^{n+1} \alpha_{ik}\right)\delta_{ij} + c_A \frac{m_i}{m_{\text{tot}}}\frac{m_j}{m_{\text{tot}}} \ . \tag{2.41}$$

$$c_u = \sum_{i=1}^{N} \boldsymbol{u}_i \ . \tag{2.42}$$

Here, the α_{ij} factors are the new variational parameters of $\boldsymbol{A}$. The factors c_u and c_A describe the translational properties of the basis functions. The basis functions are translationally invariant if both c_u and c_A are zero. For c_u, this is not a problem. For c_A, we find that $\boldsymbol{A}$ is not positive definite anymore and the norm of the basis function vanishes. We will illustrate in chapter 4, how to overcome this issue.

If $\boldsymbol{A}$ and $\boldsymbol{u}$ are parametrized according to Eqs. (2.41) and (2.42), their transformed forms in terms of $\boldsymbol{U}_x$ contain a special super structure

$$\boldsymbol{A}^{(x)} = \begin{bmatrix} \mathcal{A}^{(x)} & 0 \\ 0 & c_A \end{bmatrix} \quad \text{and} \quad \boldsymbol{u}^{(x)} = \begin{bmatrix} \boldsymbol{u}^{(x)} \\ c_u \end{bmatrix} \ . \tag{2.43}$$

The optimization of the parameter set is computationally expensive and it is therefore important that the optimization scheme is reasonably efficient. Kinghorn [38] has illustrated how the work by Magnus and Neudecker [39,40] can be used to formulate the matrix derivative for the energy functional in Eq. (2.4) for ECGs. Analytical gradients can be used to implement a conjugate-gradient optimization scheme which converges the energy to the next local minimum. For ECGs with the GVR, however, the analytical gradients are costly and other variational schemes are better suited. Expressions for analytical gradients using ECG and the GVR are presented in Section 8.2.

Random optimization algorithms are generally employed if the gradient cannot be calculated or only at great computational cost. Suzuki and Varga [2,41,42] have shown that stochastic sampling is also an efficient method of optimizing ECGs with the GVR. Here, in each sampling step, one variational parameter is replaced by a new random guess and kept if the energy of the new parameter set is lower than the energy of the original parameter set. Mátyus [30] has investigated different sampling schemes for the α_{ij} parameters in the parametrization of $\boldsymbol{A}$ according to Eq. (2.41) and using a normal distribution for the sampling of $\ln(\alpha)$ resulted in the most efficient sampling scheme.

3

Relativistic Electronic Structure Theory

In this chapter, which was published in Ref. [9], we review essentials of the relativistic quantum theory of many-electron systems in the external field of atomic nuclei, which provides the grounds for relativistic quantum chemistry [7]. This quantum theory is based on Einstein's relativity principle, i.e., on the two postulates that (i) the speed of light has the same constant value for any observer and that (ii) the mathematical form of fundamental physical laws must be the same in all frames of reference. However, computational considerations force one to sacrifice the latter of these principles. Nevertheless, methods of relativistic quantum chemistry turned out to yield accurate results for molecules containing heavy atoms and for high-resolution spectroscopy. While many reviews of the field have been published in recent years (see, e.g., Refs. [10,43–54]), we focus in this account on the core principles of this first-quantized, semi-classical theory and provide an overview of the computational obstacles that one faces when turning the theory into a practical approach for actual calculations.

The physical theory that describes the motion of electrons and photons is quantum electrodynamics (QED). It is a second-quantized theory which, besides the matter field, also treats the radiation field as quantized. Although only few-electron systems (typically two- or three-electron atoms) have been studied in this framework, QED can be considered to be the fundamental theory of chemistry. However, it turned out that the quantization of the radiation field is an unnecessary burden for almost all chemical applications.

In other words, the explicit description of particles of light, i.e., photons is almost never necessary and classical electromagnetic fields can be used instead. Hence, this theory is first-quantized as it considers only the matter field as being quantized.

Electromagnetic fields may induce a change of the state of a molecule. The photophysics of such a process can be described either by quantum dynamics, i.e., by solving the time-dependent Schrödinger equation, or by time-dependent perturbation theory (Fermi's golden rule). However, most photochemical processes simply require knowledge about the initial and final states rather than about the details of the transition from one state to the other. Accordingly, the solution of the stationary Schrödinger equation for a many-electron system is usually sufficient for the study of electronic effects in molecules.

Relativistic effects were considered of little importance in such a theory as molecular physics and chemistry belong to the realm of low-energy physics. In the 1970s this assumption was proven wrong for molecules containing heavy atoms (i.e., those with a high nuclear charge number Z). Hence, Schrödinger quantum mechanics, as a non-relativistic theory, is not sufficient for the whole of chemistry. The development of a first-quantized theory, which can account for all relativistic effects needed to understand a given problem, is required. As for QED, this theory is advised to settle on the Dirac theory of the electron, although its development is not as straightforward as one might think. However, many-electron Schrödinger quantum mechanics, which is easier to formulate, can always serve as a guiding principle — after all, almost all of organic and bio-chemistry can be described by non-relativistic Schrödinger quantum mechanics. Accordingly, in the past five decades much work has been devoted to establish a relativistic analogue based on the Dirac Hamiltonian. Here, we shall present its ingredients and discuss its pathologies as well as their remedies.

3.1 Dirac's Theory of the Electron

The one-electron Dirac Hamiltonian [55,56] $\boldsymbol{h}_D$ provides a relativistic description of a single electron in an external (classical) electromagnetic potential $\boldsymbol{V}$

$$\boldsymbol{h}_D = \left(c\boldsymbol{\alpha}\cdot\boldsymbol{p} + \boldsymbol{\beta}mc^2 + \boldsymbol{V}\right), \tag{3.1}$$

where c is the speed of light, m is the rest mass of the electron, and $\boldsymbol{p} = (p_x, p_y, p_z)^{\mathrm{T}}$ is the momentum operator. The 4×4 complex matrices

$\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3)$ and $\boldsymbol{\beta}$ are called Dirac matrices. They are uniquely determined by commutation relations, but several different representations exist. The most common choice is

$$\boldsymbol{\alpha}_i = \begin{bmatrix} \mathbf{0} & \boldsymbol{\sigma}_i \\ \boldsymbol{\sigma}_i & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{bmatrix} \mathbf{1}_2 & \mathbf{0} \\ \mathbf{0} & -\mathbf{1}_2 \end{bmatrix} \tag{3.2}$$

where $\boldsymbol{\sigma}_i$ are the three Pauli spin matrices

$$\boldsymbol{\sigma}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{\sigma}_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\sigma}_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{3.3}$$

This choice is known as the standard representation of the Dirac matrices.

The wave function of a single electron $\boldsymbol{\psi}(\boldsymbol{r})$ is an eigenfunction of $\boldsymbol{h}_D$ and thus a four-component vector matching the dimension of the Hamiltonian, called a 4-spinor. It is convenient to transfer the 2×2 block structure of the Dirac matrices to the wave function,

$$\boldsymbol{\psi}(\boldsymbol{r}) = \begin{bmatrix} \boldsymbol{\psi}^l(\boldsymbol{r}) \\ \boldsymbol{\psi}^s(\boldsymbol{r}) \end{bmatrix}. \tag{3.4}$$

The two components are known as the large (superscript "l") and the small (superscript "s") components (sometimes referred to as "upper" and "lower" components). Both, the large and the small component, are 2-spinors. For a detailed discussion of the mathematical properties of the one-electron Dirac Hamiltonian, the reader is referred to the review by Esteban, Lewin and Séré [57].

If the potential-energy operator $\boldsymbol{V}$ is spherically symmetric, $\boldsymbol{h}_D$ commutes with the total angular momentum operators $\boldsymbol{j}^2/\boldsymbol{j}_z$. However, $\boldsymbol{h}_D$ does not commute with either the orbital angular momentum operators $\boldsymbol{l}^2/l_z$ or the spin operators $\boldsymbol{s}/\boldsymbol{s}_z$. Thus, $\boldsymbol{\psi}(\boldsymbol{r})$ is not an eigenfunction of either pair of operators. One can show that for the total angular momentum operator defined as

$$\boldsymbol{j} = \boldsymbol{l}\mathbf{1}_4 + \frac{\hbar}{2}\boldsymbol{\sigma} \otimes \mathbf{1}_2, \tag{3.5}$$

with $\boldsymbol{\sigma}$ being the vector of Pauli spin matrices. $\boldsymbol{j}^2$ and $\boldsymbol{j}_i$, with $i \in (x, y, z)$, both commute with $\boldsymbol{h}^D$.

The eigenvalue spectrum of the one-electron Dirac Hamiltonian differs significantly from the non-relativistic one-electron Schrödinger Hamiltonian. Figure 3.1 illustrates the spectra of both. The one of the Schrödinger Hamiltonian consists of two parts. First, the positive continuum describing the unbound

states of the electron, and second, the discrete (quantized) bound states. The spectrum is therefore bounded from below, and the ground-state energy is well-defined as the global minimum of the energy spectrum. By contrast, the spectrum of the one-electron Dirac Hamiltonian features three parts (we follow the notation by Pestka *et al.* for the denomination of the individual parts of the spectrum [58]). First, there is the positive continuum of states $\Sigma^{(+)}$ ranging from $+mc^2$ to $+\infty$. Then, we find the bound states $E^{(1)}$ between $-mc^2$ and $+mc^2$ (for small nuclear charge numbers $E^{(1)}$ are close to $+mc^2$) and finally a negative continuum $\Sigma^{(-)}$ below $-mc^2$. $\Sigma^{(-)}$ is a set of negative-energy states (sometimes called positronic states) spanning the energy range $(-\infty, -mc^2)$. As a consequence, the Dirac Hamiltonian is not bounded from below. The negative-energy continuum states pose conceptual and practical difficulties. However, they also led to the discovery of the anti-electron (the positron), although this discovery eventually required the introduction of a new theory, namely quantum electrodynamics, in which these positrons feature positive energies and a ground state is well-defined.

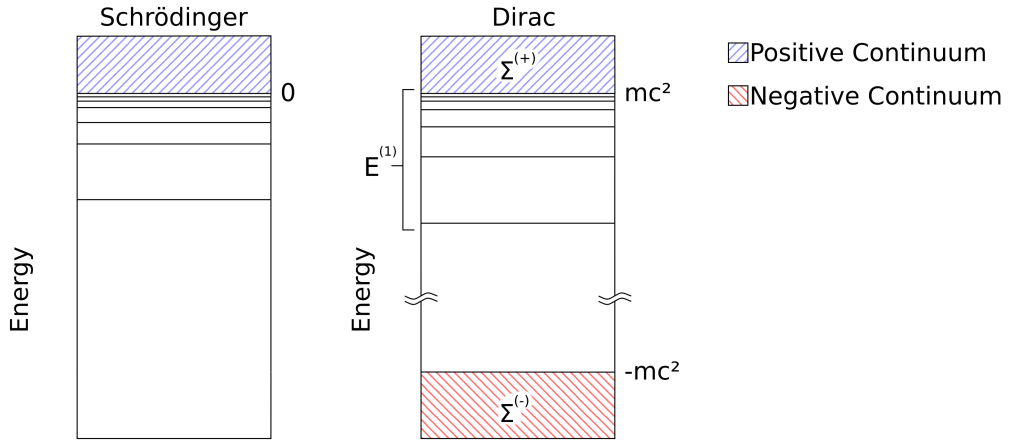

Figure 3.1: Energy eigenvalue spectra of a Schrödinger (left) and a Dirac (right) electron in an attractive external potential. Continuum states are represented by shaded areas and bound states by solid lines. Note the different zero-energy references: In the Schrödinger spectrum, the electron at rest has zero energy, while the rest energy is mc^2 for the Dirac electron.

If we compare the bound states of the Schrödinger Hamiltonian and the corresponding ones of the Dirac Hamiltonian, we see that the energies are lowered in the Dirac spectrum. This is due to kinematic relativistic effects, which are also called scalar-relativistic effects as the lowering of the energy can also be observed for quasi-relativistic Hamiltonians, in which the spin degrees of freedom have been separated and omitted. Moreover, some of

the non-relativistic bound states are split in the Dirac case, which is due to the coupling of spin and orbital angular momentum. Spin–orbit coupling is implicitly contained in the Dirac Hamiltonian (by contrast to the Schrödinger Hamiltonian, which does not depend on the Pauli spin matrices).

3.2 The Relativistic N-Electron Case

An explicit expression for a truly relativistic, first-quantized many-electron Hamiltonian is not known [59]. For a fixed number of N electrons in the external electrostatic potential of atomic nuclei, it is possible to construct an approximate model Hamiltonian $\boldsymbol{H}^{(N+)}$,

$$\boldsymbol{H}^{(N+)} = \Lambda^{(+)}\boldsymbol{H}^{(N)}\Lambda^{(+)} = \Lambda^{(+)}\boldsymbol{H}_D^{(N)}\Lambda^{(+)} + \sum_{i<j}^{N}\Lambda^{(+)}\boldsymbol{g}(i,j)\Lambda^{(+)} \ , \tag{3.6}$$

where the two-electron operator $\boldsymbol{g}(i,j)$ describes the interaction of the electrons i and j. $\boldsymbol{H}_D^{(N)}$ collects the one-electron energy contributions,

$$\boldsymbol{H}_D^{(N)} = \sum_{i=1}^{N}\boldsymbol{h}_D(i), \tag{3.7}$$

with

$$\boldsymbol{h}_D(i) = \mathbf{1}_4(1)\otimes\cdots\otimes\mathbf{1}_4(i-1)\otimes\boldsymbol{h}_D\otimes\mathbf{1}_4(i+1)\otimes\cdots\otimes\mathbf{1}_4(N). \tag{3.8}$$

The many-electron Hamiltonian for N electrons thus has a tensor structure of dimension 4^N. Although based on the Dirac Hamiltonian it possesses a well-defined ground state by virtue of projection operators. The projection operators $\Lambda^{(+)}$ must be defined in such a way that only the positive-energy bound and continuum electronic states are accessible, i.e., all positronic states are dismissed. The problem is that these projection operators are not known a priory since they depend on the electronic solutions of $\boldsymbol{H}^{(N+)}$. Approximations based on quantum-electrodynamical arguments have been proposed [60–65] and aspects involving practical calculations were studied, for example, by Indelicato [66,67]. Recently, these issues have again attracted attention [68–70]. For the sake of brevity, we omit the projection operators in the following expressions.

The interaction operator $\boldsymbol{g}(i,j)$ for two electrons i and j in Eq. (3.6) can be approximated by the electrostatic Coulomb operator (in Hartree atomic units)

$$\boldsymbol{g}_C(i,j) = \frac{1}{|\boldsymbol{r}_{ij}|}, \tag{3.9}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ is the inter-electronic distance vector. The resulting many-electron Hamiltonian defines the Dirac–Coulomb model. This electrostatic interaction omits magnetic interactions and assumes that the interaction is transmitted instantaneously. Since the speed of light c is the upper limit at which interactions between electrons are transmitted by photons, the retardation of the interaction between the two electrons should be taken into account. Retardation and magnetic effects can be approximately described to lowest order by the frequency-independent Breit-operator,

$$\boldsymbol{g}_B(i,j) = -\frac{1}{2}\left(\frac{\boldsymbol{\alpha}_i \cdot \boldsymbol{\alpha}_j}{|\boldsymbol{r}_{ij}|} + \frac{(\boldsymbol{r}_{ij} \cdot \boldsymbol{\alpha}_i)(\boldsymbol{r}_{ij} \cdot \boldsymbol{\alpha}_j)}{|\boldsymbol{r}_{ij}|^3}\right), \tag{3.10}$$

to be added to $g_C(i,j)$.

The non-relativistic limit of the Hamiltonian is formally obtained in the limit of an infinite speed of light

$$\boldsymbol{H}_{\mathrm{NR}}^{(N)} = \lim_{c\to\infty}\left[\boldsymbol{H}^{(N)} - \sum_{i=1}^{N}\beta mc^2\right], \tag{3.11}$$

where the rest energy of all electrons has been subtracted to match the non-relativistic energy scale. The resulting Hamiltonian yields the one-component electronic Schrödinger Hamiltonian

$$\boldsymbol{H}_{\mathrm{NR}}^{(N)} \to \sum_{i=1}^{N}\left[\frac{\boldsymbol{p}_i^2}{2m} + \boldsymbol{V}_{\mathrm{ext},i}\right] + \sum_{i<j}\tilde{g}(i,j), \tag{3.12}$$

where $\boldsymbol{V}_{\mathrm{ext},i}$ collects all interactions of electron i with external electromagnetic fields (e.g., with the electrostatic field of the atomic nuclei in BO approximation) and where $\tilde{g}(i,j)$ is the electron–electron interaction operator connected to its relativistic analogue $g(i,j)$ by a unitary transformation (see below).

The relativistic N-electron Hamiltonian can also be described in terms of its model space $\mathcal{H}^{(N)}$ [71]. Accordingly, $\mathcal{H}^{(N)}$ can be decomposed into its one-electron subspaces

$$\mathcal{H}^{(N)} = \left[(\mathcal{H}^{(1)})^{\otimes N}\right]^{\mathrm{A}}, \tag{3.13}$$

where the superscript A indicates antisymmetry with respect to pairwise exchange of electrons. The one-electron model spaces $\mathcal{H}^{(1)}$ can be further decomposed into

$$\mathcal{H}^{(1)} = \mathcal{H}^{(l)} \otimes \mathcal{H}^{(s)}, \tag{3.14}$$

where $\mathcal{H}^{(l)}$ is the model space of the large component and $\mathcal{H}^{(s)}$ is the model space of the small component [71, 72].

To understand the spectrum of the N-electron Hamiltonian, we consider the simplest case of two electrons, i.e., $N = 2$, first. Figure 3.2 illustrates the spectrum of this Hamiltonian under the assumption that the two electrons are *not interacting*. We can identify six different parts according to the pairing of the one-electron Dirac energies of the individual electrons (again, we follow the notation of Pestka *et al.* [58]). First, there are the discrete bound states denoted by $E^{(2)} = E_1^{(1)} + E_2^{(1)} \in (-2mc^2, +2mc^2)$ where $E_1^{(1)}$ and $E_2^{(1)}$ are the bound states of Figure 3.1. Observing that Dirac one-electron states may also be continuum states for both of the two electrons, we can identify three additional sets of states: the positive continuum $\Sigma^{(++)}$ spans the range $(+2mc^2, +\infty)$, whereas the negative continuum $\Sigma^{(--)}$ spans the range $(-\infty, -2mc^2)$. A third continuum is generated by a combination of negative-energy and positive-energy continuum states. This so-called Brown–Ravenhall continuum $\Sigma^{(+-)}$ covers the complete energy range, i.e., $(-\infty, +\infty)$. It is the reason for the non-existence of a ground state for the fully interacting two-electron system. The final two sets are mixtures of bound and continuum states. Here, one electron is in a bound state, while the other electron is in either a positive- or negative-energy continuum state: The former set is denoted as $\Sigma_d^{(+)}$ and spans the range $(E^{(1)} + mc^2, +\infty)$, while the latter is denoted as $\Sigma_d^{(-)}$ and spans the range $(-\infty, E^{(1)} - mc^2)$. If the electron-electron interaction is now switched on, we either face continuum or autoionizing states. In the autoionizing states, a bound state couples to the continuum, which would lead to its decay. As a consequence, the Dirac–Coulomb model (without projection) is not considered a useful physical Hamiltonian. However, it is the most widely applied Hamiltonian as projection on the square-integrable one-electron bound states yields remarkably accurate results, despite their dependence on the choice of the projection operators (see below for a further discussion of these issues).

It is important to understand that the preceding analysis is based on the mathematically possible combinations of the one-electron Dirac energies in the decoupled problem. Dirac already noted that the existence of the negative-energy states requires a physical solution and hence proposed to occupy all the infinitely many states by electrons, which would go unnoticed by us if the potential created is homogeneous in space. Electron–positron pair creation is then an excitation process of an electron from the negative-energy continuum, the so-called Dirac sea, which leaves a (positively charged) hole, the anti-electron, behind. This process shows a principle that is also preserved by QED, namely that the total charge is conserved by such processes and thus a constant of the motion. Clearly, the infinitely many negative charges in these states create new conceptual problems as the theory was designed to describe

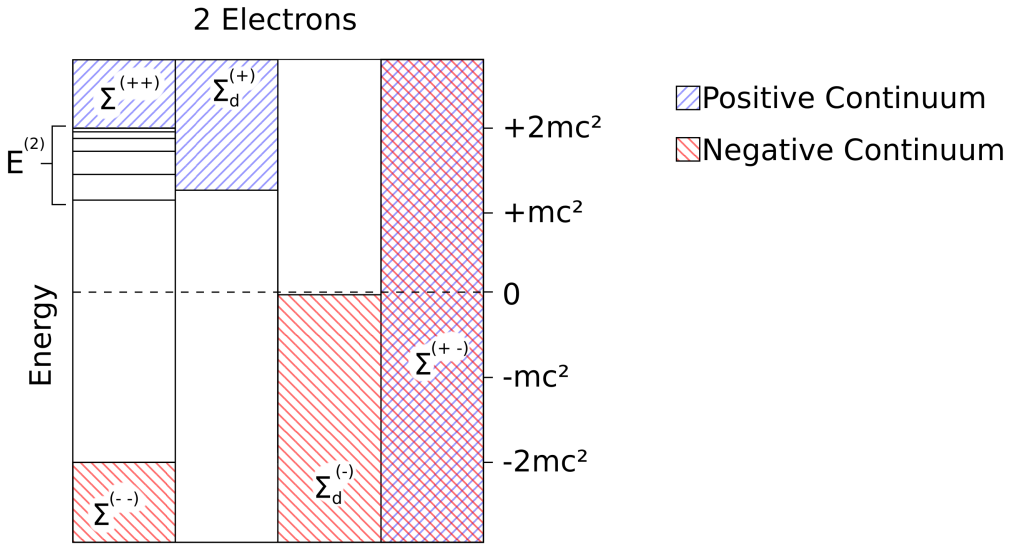

Figure 3.2: Schematic representation of the spectrum of a system of two *non-interacting* Dirac electrons. Continuum states are represented by shaded areas and bound states by solid lines. The orientation of the lines in the shaded areas indicates (in addition to the color) whether the continuum has positive- or negative-energy contributions. Note that the bound states will be found close to $+2mc^2$ for low nuclear charge numbers, but may approach $-2mc^2$ for very high nuclear charges (i.e., $Z \approx 137$ for point-like nuclei or even $Z \approx 170$ for finite-sized nuclei).

only a single electron. In QED, the issue is resolved by letting these states being accessible for anti-electrons with positive energies by virtue of normal ordering. Here, we explicitly aim for a first-quantized theory and have to deal with the negative-energy states. Especially in basis-set expansion approaches (see below), they cannot be simply neglected, as this would cause basis-set incompleteness issues — especially for molecular property calculations, for which they must not be neglected (e.g., magnetic properties).

3.3 Approximation of the State Function

The N-electron state function $\mathbf{\Psi}$ can be approximated in various ways. Here, we focus on analytic basis-set expansions, in which a model space is constructed from a finite number of n basis functions, rather than on numerical grid-based methods. The state function is then obtained as a linear combination of many-electron basis functions within this model space,

$$\mathbf{\Psi} = \sum_{I=1}^{n} C_I \mathbf{\Phi}_I, \tag{3.15}$$

(in case of a complete basis set: $n = \infty$). The basis-set expansion will, however, not be as straightforward as in the non-relativistic theory, which can be understood by factorizing the model space $\mathcal{H}$ with respect to the components of the state function. For $N = 2$ the factorization may be written as [58]:

$$\begin{aligned}
\mathcal{H}^{(ll)} &= \left[\mathcal{H}_1^l \otimes \mathcal{H}_2^l\right]^{\mathrm{A}} \ , \\
\mathcal{H}^{([ls])} &= \left[\mathcal{H}_1^l \otimes \mathcal{H}_2^s \oplus \mathcal{H}_1^s \otimes \mathcal{H}_2^l\right]^{\mathrm{A}} , \\
\mathcal{H}^{(ss)} &= [\mathcal{H}_1^s \otimes \mathcal{H}_2^s]^{\mathrm{A}} \ . \qquad (3.16)
\end{aligned}$$

The individual model spaces can then be expanded analogously to Eq. (3.15). The basis functions are functions of all electronic coordinates, $\overline{\boldsymbol{r}} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)$. The approximated state function $\boldsymbol{\Psi}$ has to fulfill the Pauli antisymmetry principle, which is the reason for the $'\mathrm{A}'$ superscript in the equations above. The antisymmetry may be explicitly written for the two-electron case, $N = 2$, as [58],

$$\begin{aligned}
\boldsymbol{\Psi}^{(ll)}(\boldsymbol{r}_1, \boldsymbol{r}_2) &= -\boldsymbol{\Psi}^{(ll)}(\boldsymbol{r}_2, \boldsymbol{r}_1) \ , \\
\boldsymbol{\Psi}^{(ls)}(\boldsymbol{r}_1, \boldsymbol{r}_2) &= -\boldsymbol{\Psi}^{(sl)}(\boldsymbol{r}_2, \boldsymbol{r}_1) \ , \\
\boldsymbol{\Psi}^{(ss)}(\boldsymbol{r}_1, \boldsymbol{r}_2) &= -\boldsymbol{\Psi}^{(ss)}(\boldsymbol{r}_2, \boldsymbol{r}_1) \ . \qquad (3.17)
\end{aligned}$$

Basis functions $\boldsymbol{\Phi}_I$, which also must fulfill the Pauli principle, can be constructed in a multitude of ways. Commonly, the N-electron basis functions are constructed from one-electron functions, i.e., from 4-spinor basis states. It is, however, possible to include two-electron functions already from the outset. Such functions are convenient to describe the electronic cusp that emerges at the coalescence point of two interacting electrons, and they are known as correlation factors in quantum chemistry. N-electron basis functions that contain such a dependence on inter-electronic distances are called explicitly correlated basis functions. Since they have become popular in non-relativistic quantum chemistry only recently (due to technical advances in the evaluation of the integrals), their benefits in relativistic calculations yet have to be exploited, although they may cause severe stability problems in relativistic calculations (see next section).

In principle, the N-electron basis functions $\boldsymbol{\Phi}_I$ can be constructed directly, exploiting knowledge about the analytic form of the one-electron state functions. However, there is no universal definition for such a function. In the original work by Hylleraas and Undheim [73, 74] on the non-relativistic helium

problem, the state function was directly approximated by

$$\Psi_{\mathrm{Hyl}}^{(\{c_{nlm}\})}(\boldsymbol{r}_1,\boldsymbol{r}_2,\boldsymbol{r}_{12}) = \exp\left(-\frac{1}{2}(\boldsymbol{r}_1+\boldsymbol{r}_2)\right) \times \sum_{nlm=0}^{\infty} c_{nlm}(\boldsymbol{r}_1+\boldsymbol{r}_2)^n(\boldsymbol{r}_1-\boldsymbol{r}_2)^{2l}(\boldsymbol{r}_{12})^m, \tag{3.18}$$

where c_{nlm} are parameters to be optimized. Various studies have expanded on this ansatz (see, e.g., Refs. [75–78]). In the relativistic realm, basis functions constructed in the spirit of the Hylleraas state function have been successfully applied by Pestka *et al.* (see Ref. [58] and references therein).

Another choice for basis functions are Gaussian-type functions, which allow for comparatively easy analytic integral evaluation, while preserving some of the structure of the one-electron state functions. In this case, a many-electron basis function can be written directly in a compact form for an N-electron system:

$$\Phi_I(\overline{\boldsymbol{r}}) = \theta(\overline{\boldsymbol{r}})\exp\left(-\frac{1}{2}\overline{\boldsymbol{r}}^{\mathrm{T}}\boldsymbol{A}^{(I)}\overline{\boldsymbol{r}}\right). \tag{3.19}$$

Recall that $\overline{\boldsymbol{r}}$ collects *all* electronic coordinates. The matrix $\boldsymbol{A}^{(I)}$ is a positive definite $N\times N$ matrix containing all variational parameters of this many-electron basis function. Note that it explicitly depends on the inter-electronic distances $\boldsymbol{r}_{ij}$ if $\boldsymbol{A}^{(I)}$ is a full square matrix. Such ECGs were introduced by Boys [79] and Singer [80] into non-relativistic theory. For the simplification of $\boldsymbol{A}^{(I)}$ being a diagonal matrix, a decoupled basis function results,

$$\boldsymbol{\Phi}_I^{\mathrm{IPM}}(\overline{\boldsymbol{r}}) = \theta(\overline{\boldsymbol{r}})\exp\left(-\frac{1}{2}\overline{\boldsymbol{r}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{diag}}^{(I)}\overline{\boldsymbol{r}}\right) = \theta(\overline{\boldsymbol{r}})\prod_{i=1}^{N}\exp\left(-\frac{A_{ii}^{(I)}}{2}\boldsymbol{r}_i^2\right), \tag{3.20}$$

in which the individual electronic coordinates are separated and thus can only describe the uncoupled problem well, often called the independent particle model (IPM). Note that, for the sake of clarity, we have omitted the anti-symmetrization step, which requires the application of an antisymmetrization operator which implements the Pauli principle.

Boys [79] and Singer [80] chose for the angular part of a non-relativistic basis function a product of polynomials

$$\theta_P(\overline{\boldsymbol{r}}) = \prod_i^N x_i^{l_i} y_i^{m_i} z_i^{n_i}, \tag{3.21}$$

while Suzuki and Varga developed a GVR of the eigenfunctions of the $\boldsymbol{L}^2/L_z$ operators [2, 35, 36]

$$\theta_{\mathrm{GVR}}(\overline{\boldsymbol{r}}) = |\boldsymbol{v}|^{2K+L}Y_{LM_L}(\boldsymbol{v})\ . \tag{3.22}$$

with $K \in \mathbb{N}_0$, the spherical harmonics Y_{LM_L} of degree L and order M_L, and the global vector

$$\boldsymbol{v} = (\boldsymbol{u} \otimes \mathbf{1}_3)\overline{\boldsymbol{r}}. \tag{3.23}$$

The GVR has been successfully applied to non-relativistic calculations [29,30, 81–83] and can, in principle, be applied for the construction of relativistic ECGs [81].

While we have given possible expressions for the many-electron basis functions Φ_I above, the standard procedure is to construct them step-wise in order to find a basis-set expansion in Eq. (3.15) that is as compact as possible (for some desired accuracy). In a first step, $n = 1$ is chosen. Then, the IPM is adopted in order to write the only remaining basis function Φ_1 as an antisymmetrized direct product of N one-electron functions $\boldsymbol{\psi}_i(\boldsymbol{r})$,

$$\boldsymbol{\Phi}_1(\overline{\boldsymbol{r}}) = \mathrm{A}\boldsymbol{\psi}_1(\boldsymbol{r}_1) \otimes \cdots \otimes \boldsymbol{\psi}_N(\boldsymbol{r}_N), \tag{3.24}$$

with A being an antisymmetrization operator, which explicitly implements the Pauli principle (i.e., it produces all permutations of electronic coordinates and changes the sign per pair permutation of coordinates). The ansatz $\boldsymbol{\Psi} \approx \boldsymbol{\Phi}_1$ defines the Dirac–Hartree–Fock model, i.e., the relativistic analogue of Hartree–Fock theory. The individual one-electron functions $\boldsymbol{\psi}_i(\boldsymbol{r})$, which are 4-spinors in the relativistic theory, are expanded in terms of one-electron basis functions,

$$\boldsymbol{\psi}_i(\boldsymbol{r}) = \sum_{j=1}^{n_i} c_j^{(i)} \boldsymbol{\phi}_j(\boldsymbol{r}), \tag{3.25}$$

where the spinor coefficients $c_j^{(i)}$ are the parameters to be determined by a variational procedure (see below). Methods based on such basis functions are called four-component methods. The two most common choices for a component of the (four-component) basis spinors $\boldsymbol{\phi}_j$ are Slater-type orbitals (STO) [84]

$$\phi_j^{(\mathrm{STO})}(\boldsymbol{r}) = \theta(\boldsymbol{r}) \exp\left(-\zeta_j |\boldsymbol{r}|\right) \tag{3.26}$$

and Gaussian-type orbitals (GTO) [37]

$$\phi_j^{(\mathrm{GTO})}(\boldsymbol{r}) = \theta(\boldsymbol{r}) \exp\left(-\zeta_j \boldsymbol{r}^2\right), \tag{3.27}$$

where $\zeta_j > 0$ are exponents to be suitably chosen and $\theta(\boldsymbol{r})$ is the angular part of the function which is related to spherical harmonics. Boys [37] introduced

$$\theta(\boldsymbol{r}) = x^l y^m z^n \tag{3.28}$$

to describe this angular part in non-relativistic calculations (Cartesian GTOs), However, it should be noted that the most efficient way to expand a 4-spinor

according to Eq. (3.25) is to exploit the 2×2 super-structure of the Dirac Hamiltonian and thus to have one GTO (or STO) for each, the large and small, component of the 4-spinor. Then, $\theta(\boldsymbol{r})$ can be chosen as a Pauli 2-spinor which is an eigenfunction of the spin–orbit coupling operator and thus constructed as a linear combination of products of spin and orbital angular momentum eigenfunctions multiplied by a Clebsch–Gordan coefficient.

In recent years it has become increasingly obvious that the number of one-electron and thus many-electron basis functions can only be reduced if the inter-electronic distance is explicitly taken into account. Hence, explicitly correlated basis functions are employed in non-relativistic quantum chemistry to make correlated methods — such as configuration interaction (CI), coupled cluster (CC), and Møller–Plesset (MP) perturbation theory — more efficient with respect to the number of one-electron basis functions needed for a desired accuracy. Several variants have been analyzed in the non-relativistic framework and the most efficient of them could be considered also for relativistic calculations. While the old R_{12} methods directly rely on the inter-electronic distances as factors, F_{12} methods use a function of the inter-electronic distance. The form of the function is determined by computational efficiency and accuracy considerations but often reminiscent of Slater- or Gaussian-type functions (see Refs. [85, 86] for reviews). Their advent in a straightforward application to relativistic four-component quantum-chemical methods may be hampered by the unboundedness of the (unprojected) Dirac–Coulomb Hamiltonian, as we discuss in the following.

3.4 Variational Approaches

The state function of the ground state of the (projected) relativistic many-electron Hamiltonian $\boldsymbol{H}^{(N)}$ is a critical point of the Rayleigh quotient

$$\mathrm{E}(\boldsymbol{H}^{(N)}, \boldsymbol{\Psi}) = \frac{\langle \boldsymbol{\Psi} | \, \boldsymbol{H}^{(N)} \, | \boldsymbol{\Psi} \rangle}{\langle \boldsymbol{\Psi} \, | \boldsymbol{\Psi} \rangle}. \tag{3.29}$$

3.4.1 Variational Collapse

The critical point can be obtained by minimization of $\mathrm{E}(\boldsymbol{H}^{(N)}, \boldsymbol{\Psi})$ with respect to the parameters in $\boldsymbol{\Psi}$. Depending on the ansatz in Eq. (3.15), excited states, orthogonal to the ground state, may be obtained as well. In practice, the relativistic variational approach is reminiscent to the one in non-relativistic theory and thus similar methods have been developed [87–89]. It must be emphasized, though, that they all require some sort of projection onto the

positive-energy states as the unboundedness of $\boldsymbol{H}^{(N)}$ would otherwise create a collapse of any variational procedure yielding dramatically negative and totally artificial electronic energies. Most of these projections are implicit by making sure that the one-electron basis functions fulfill the kinetic-balance condition, i.e., the relation between large and small components of the Dirac 4-spinor (see Refs. [23–25] for recent discussions and for further references cited therein) and that only square-integrable, i.e., bound-state solutions are obtained. One may show that the lowest-energy bound state is a saddle point of $E(\boldsymbol{H}^{(N)}, \boldsymbol{\Psi})$ with respect to the extremalization of the corresponding large and small components in $\boldsymbol{\Psi}$ [90–92]. Apart from the variational collapse, the expansion of the state function in terms of a finite one-electron basis set can lead to approximate energies which are close to the complete-basis-set result but *lower*, which has been called *prolapse* by Fægri [93]. Subsection 3.4.3 focuses on this problem by presenting its origin and possible solutions. Furthermore, if explicitly correlated basis functions are used the Brown–Ravenhall continuum causes variational issues because of the energetical overlap of the set of bound states and $\Sigma^{(+-)}$. Subsection 3.4.5 focuses on this problem and reviews a possible solution: the Complex Coordinate Rotation (CCR) method. For the next subsection, we assume that the Dirac–Coulomb model with projection is applied.

3.4.2 Four-Component Methods

If the state function is minimized with respect to the expansion parameters of Eq. (3.15), the resulting methods are called CI or exact-diagonalization methods. There are various flavours of CI methods depending on the types of function from which the model space is constructed (see [94] for an overview in the non-relativistic case). Minimization of $\mathrm{E}(\boldsymbol{H}^{(N)}, \boldsymbol{\Psi}_{\mathrm{CI}})$ with respect to the C_I expansion parameters yields the general eigenproblem

$$\boldsymbol{HC} = \boldsymbol{SCE}, \tag{3.30}$$

where $\boldsymbol{H}$ is an $n \times n$ matrix with elements $\boldsymbol{H}_{IJ} = \langle \boldsymbol{\phi}_I | \, \boldsymbol{H}^{(N)} \, | \boldsymbol{\phi}_J \rangle$. The matrix $\boldsymbol{C}$ collects the eigenvectors of $\boldsymbol{H}$ containing the C_I expansion parameters of Eq. (3.15) for different electronic states (note that the state index had not explicitly been denoted in all previous equations). $\boldsymbol{S}$ is an $n \times n$ overlap matrix which reduces to the n-dimensional unit matrix for orhogonal basis functions Φ_I. The elements of this metric are calculated by $\boldsymbol{S}_{IJ} = \langle \boldsymbol{\Phi}_I \, | \boldsymbol{\Phi}_J \rangle$. $\boldsymbol{E}$ is an $n \times n$ diagonal matrix containing the energies of the electronic states.

Minimization of $\mathrm{E}(\boldsymbol{H}^{(N)}, \boldsymbol{\Phi}_1)$ with respect to the expansion parameters of the one-electron basis functions as described in Eq. (3.25) leads to the relativistic Dirac–Hartree–Fock–Roothaan equations which we refrain from explicitly

writing down for the sake of brevity. As for non-relativistic Roothaan–Hall equations [95, 96], the coefficient functions in these equations depend on their solution and hence they have to be solved iteratively resulting in a self-consisting field (SCF) algorithm. The interaction of the electrons is treated in an averaged way in the IPM. In order to improve on the IPM, the approximation for the state function can be extended in post-DHF methods, called correlation methods, by increasing the number of many-electron basis functions in Eq. (3.15). Existing correlation methods, for which we provide only a few selected references for the sake of brevity, comprise MP perturbation theory (non-relativistic: [97,98]; relativistic: [99–101]), CI (non-relativistic.: [102–104]; relativistic.: [105–108]), multi-configuration SCF (MCSCF) (non-relativistic Refs.: [109–111]; relativistic Refs.: [66, 112–116]) including the complete active space SCF (CASSCF) variant (non-relativistic: [117,118]; relativistic: [119]), and CC (non-relativistic: [120–123]; relativistic: [124–127]).

One aspect of avoiding variational collapse is based on relating the one-electron model spaces $\mathcal{H}^l$ and $\mathcal{H}^s$ to one another [14–18] and constructing the basis functions accordingly. From the structure of the one-electron Dirac Hamiltonian, one can show that the two model spaces are related by [18]

$$\frac{c\boldsymbol{\sigma}\cdot\boldsymbol{p}}{2mc^2 - V + E}\mathcal{H}^l \subset \mathcal{H}^s \ . \tag{3.31}$$

Noting that $2mc^2 \gg E - V$ for molecules, we see that this relation can be approximately fulfilled as

$$\frac{\boldsymbol{\sigma}\cdot\boldsymbol{p}}{2mc}\mathcal{H}^l \subset \mathcal{H}^s \ . \tag{3.32}$$

which is the (approximate) kinetic-balance condition [15]. Basis functions that fulfill the kinetic-balance condition must be used to construct 4-spinors. In the same way as one can relate $\mathcal{H}^l$ to $\mathcal{H}^s$, the opposite is also possible. Basis sets constructed from both relations are called dual basis sets [23, 128] and are considered to show better variational stability and faster and smoother convergence [25]. Note that the relation in Eq. (3.32) has been derived in terms of the one-electron model spaces. Therefore, they can only be applied to basis sets generated from one-electron functions such as STO or GTO. For explicitly correlated basis function no simple relations exist [18, 58, 72].

3.4.3 Prolapse

The origin of prolapse can be studied already for a single electron. For an exact representation of its one-electron state function in a complete basis, the identity [90]

$$[\boldsymbol{\sigma}\cdot\boldsymbol{p}]^2 = [\boldsymbol{p}^2] \tag{3.33}$$

holds, where the square brackets denote the matrix representation of the operators. This relation is associated with the non-relativistic limit discussed in section 3.1. In a finite basis set, the identity is not fulfilled anymore, but the following relation holds [129, 130]:

$$[\boldsymbol{\sigma} \cdot \boldsymbol{p}]^2_{ii} < [\boldsymbol{p}^2]_{ii} = -2m[T]_{ii}. \tag{3.34}$$

This effect has been called the finite-basis disease [90] and can be related to the (non-relativistic) kinetic energy T.

3.4.4 Two-Component Methods

For molecular physics and chemistry, the creation of electron–positron pairs is not an energetically feasible process as it would require an energy of at least $2mc^2$. Hence, a relativistic first-quantized theory for molecular sciences does not need to account for this process. Accordingly, the negative-energy continuum states could already be eliminated at the level of the one-electron Hamiltonian so that all pathologies known for the four-component methods can be eliminated from the outset. A block diagonalization of the Dirac Hamiltonian can achieve this goal and yields decoupled positive- and negative-energy states so that one may focus on the block of the Hamiltonian that yields the positive-energy states. The corresponding state function then features only two components, which is the reason for the name of these methods.

The block diagonalization can be achieved by a suitable unitary transformation $\boldsymbol{U}$ of the Dirac Hamiltonian $\boldsymbol{h}_D$,

$$\boldsymbol{U}\boldsymbol{h}_D\boldsymbol{U}^\dagger = \begin{bmatrix} \boldsymbol{h}_{D+} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{h}_{D-} \end{bmatrix} \quad \text{and} \quad \boldsymbol{U}\boldsymbol{\psi} = \begin{bmatrix} \tilde{\boldsymbol{\psi}}^l \\ \boldsymbol{0} \end{bmatrix}. \tag{3.35}$$

There exist many different possibilities for choosing $\boldsymbol{U}$, which can be grouped into three families. First, there are sequential approaches, which expand the transformation $\boldsymbol{U}$ in a series of unitary transformations:

$$\boldsymbol{U} = \ldots \boldsymbol{U}_3\boldsymbol{U}_2\boldsymbol{U}_1\boldsymbol{U}_0 \ . \tag{3.36}$$

The only variationally stable one among these is based on a transformation by Douglas and Kroll [131] and was developed into a working method by Hess [132]. It expands $\boldsymbol{h}_D$ in terms of the potential $\boldsymbol{V}$ and can be conveniently evaluated in a one-electron basis set that diagonalizes the squared momentum operator, i.e., the matrix $\boldsymbol{p}^2$. This method is called the Douglas–Kroll–Hess (DKH) method [47, 49, 133] and the transformed Hamiltonian is known as the Douglas–Kroll–Hess Hamiltonian (of some order in $\boldsymbol{V}$). The exact DKH

transformation requires an infinite number of decoupling steps [134,135]. In practical calculations, the transformation is truncated after some sufficiently high order. For valence properties, second order is often sufficient, but fourth order is recommended [136,137]. For core properties, like the contact density [138–141], higher orders are required, which can be efficiently derived in practical calculations [142–144]. Siedentop and Stockmeyer have studied the convergence behaviour of DKH Hamiltonians analytically [145, 146].

Furthermore, a two-step method exists which is based on work by Barysz, Sadlej, and Snijders and therefore called in our work BSS approach [147–150]. The first step of the transformation $\boldsymbol{U}_{\mathbf{BSS}}$ is the free-particle Foldy–Wouthuysen transformation [151] $\boldsymbol{U}_{\mathrm{fpFW}}$ of $\boldsymbol{h}_D$, which was inspired by the fact that this is the mandatory first transformation in the DKH protocol [134]. However, for a Dirac electron in an electrostatic potential, the fpFW unitary transformation cannot achieve block diagonalization. Exact decoupling is then achieved only after a second unitary transformation $\boldsymbol{U}_1$, i.e.,

$$\boldsymbol{U}_{\mathbf{BSS}} = \boldsymbol{U}_1 \boldsymbol{U}_{\mathrm{fpFW}} \ . \tag{3.37}$$

The second transformation can be expressed in terms of a non-hermitian operator $\boldsymbol{R}$

$$\boldsymbol{U}_1(\boldsymbol{R}) = \begin{bmatrix} (\mathbf{1}_2 + \boldsymbol{R}^\dagger \boldsymbol{R})^{-1/2} & (\mathbf{1}_2 + \boldsymbol{R}^\dagger \boldsymbol{R})^{-1/2} \boldsymbol{R}^\dagger \\ -(\mathbf{1}_2 + \boldsymbol{R}\boldsymbol{R}^\dagger)^{-1/2} \boldsymbol{R} & (\mathbf{1}_2 + \boldsymbol{R}\boldsymbol{R}^\dagger)^{-1/2} \end{bmatrix} . \tag{3.38}$$

The matrix form of $\boldsymbol{R}$ is obtained iteratively by solving the matrix equation

$$\boldsymbol{h}_{ss}\boldsymbol{R} = \boldsymbol{R}\boldsymbol{h}_{ll} + \boldsymbol{R}\boldsymbol{h}_{ls}\boldsymbol{R} - \boldsymbol{h}_{sl}. \tag{3.39}$$

The third method yields $\boldsymbol{U}$ in a single step. It is rather straightforward to show that the kinetic-balance relation

$$\boldsymbol{\psi}^s = \boldsymbol{X}\boldsymbol{\psi}^l, \tag{3.40}$$

which relates the large and small component of the 4-spinor through the action of an operator $\boldsymbol{X}$, can be used to provide an analytic expression of the unitary transformation [22],

$$\boldsymbol{U}_{\mathrm{X2C}}(\boldsymbol{X}) = \begin{bmatrix} (\mathbf{1}_2 + \boldsymbol{X}^\dagger \boldsymbol{X})^{-1/2} & (\mathbf{1}_2 + \boldsymbol{X}^\dagger \boldsymbol{X})^{-1/2} \boldsymbol{X}^\dagger \\ -(\mathbf{1}_2 + \boldsymbol{X}\boldsymbol{X}^\dagger)^{-1/2} \boldsymbol{X} & (\mathbf{1}_2 + \boldsymbol{X}\boldsymbol{X}^\dagger)^{-1/2} \end{bmatrix} . \tag{3.41}$$

Note that the above equation was also the foundation for Eq. (3.38). For a one-electron Dirac Hamiltonian neglecting electron–electron interaction operators, $\boldsymbol{X}$ can be efficiently calculated from its eigenfunctions (but requires a

diagonalization of its matrix representation in a one-electron 4-spinor basis set). Neglecting the two-electron interaction for the unitary transformation introduces a picture-change error [152] on the electron–electron interaction, which turns out to be small in most cases though. The one-step approach has become known as the exact two-component approach (X2C) [153–164].

For a direct comparison of all three unitary decoupling approaches we refer the interested reader to Ref. [48].

Another variational scheme for the elimination of the small component of the spinor is the regular approximation. Its most prominent variant is the zeroth-order regular-approximation (ZORA) [165–171]. It employs only the first term of the Taylor-series expansion that defines the regular approximation. ZORA is a computationally cheap method and has become very successful in chemistry, especially for the calculation of molecular properties [172, 173].

3.4.5 Brown–Ravenhall Disease and its Cure

For more than one electron, employing an expansion in terms of explicitly correlated basis functions without any projection tricks, the overlap between the Brown–Ravenhall continuum $\Sigma^{(+-)}$ and the bound states $E^{(N)}$ causes an arbitrary decrease of the N-electron energy $E^{(N)}$ depending on the finite basis chosen. This unphysical effect is called the Brown–Ravenhall disease [174] and is a potential source for variational collapse. Bound states coupling to a continuum are generally called resonances. Since resonances are self-decaying states with a finite life time τ and constitute a common problem in scattering theory, several methods have been developed to overcome this problem. A reliable method to treat resonances is the CCR method [175–178] (also known as Complex Scaling). The interested reader is referred to either the review by Reinhardt [179] or the one by Ho [180]; also the book on resonances by Kukulin, Krasnopol'sky and Horáček [181] contains a chapter on CCR. The CCR method is based on the transformation

$$\overline{\boldsymbol{r}} \mapsto \overline{\boldsymbol{r}} \exp(i\Theta), \tag{3.42}$$

where Θ is called the rotation angle. After this transformation, Eq. (3.6) becomes

$$\boldsymbol{H}^{(N)}(\overline{\boldsymbol{r}} \exp(i\Theta))\boldsymbol{\Psi}(\overline{\boldsymbol{r}}) = z\boldsymbol{\Psi}(\overline{\boldsymbol{r}}) \tag{3.43}$$

where z is a complex number. The real and imaginary parts of z have very distinct physical meanings. $\mathrm{Re}(z)$ is the energy of the state, whereas $\Gamma = -2\mathrm{Im}(z)$ is the width of the resonance and is related to the life time τ of the state as $\tau = \hbar/\Gamma$. Since bound states are stable and not auto-decaying in a

stationary problem, their life time is infinite and thus $\Gamma_{\mathrm{bound}} = 0$. This implies that bound states are not affected by the choice of Θ as long as $|\Theta| \leq \pi/2$. The continuum states, however, are rotated into the complex plane by an the angle Θ. Figure 3.3, drafted after the figures given in Refs. [58, 182, 183], illustrates the eigenvalue spectra of a one-electron and a two-electron system after the dilatation transformation described in Eq. (3.42) has been carried out. In the one-electron case [58], we find several bound states, marked

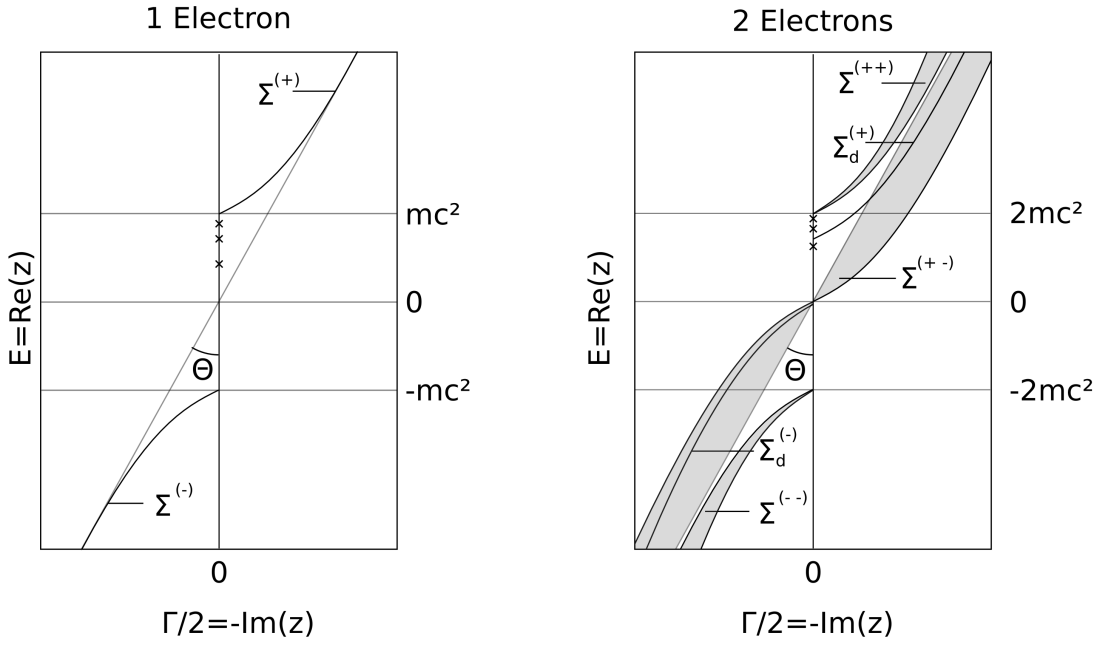

Figure 3.3: Sketch of the spectra of a one-electron and a two-electron system after the dilatation transformation in Eq. (3.42) drafted according to the figures in Refs. [58,182,183]. Bound states are marked with crosses on the real axis along the origin of the imaginary axis. Continuum states are represented by solid lines.

by crosses along the real axis on the origin of the imaginary axis. The two sets of continuum states, $\Sigma^{(+)}$ and $\Sigma^{(-)}$, are represented by solid lines starting from $\mathrm{Re}(z) = mc^2$ and $\mathrm{Re}(z) = -mc^2$. These starting points are called resonance thresholds. The two-electron case features a total of five sets of continuum states [58]. The $\Sigma^{(++)}$ and $\Sigma^{(--)}$ states have resonance thresholds of $\mathrm{Re}(z) = 2mc^2$ and $\mathrm{Re}(z) = -2mc^2$, whereas the $\Sigma_d^{(+)}$ and $\Sigma_d^{(-)}$ states have thresholds with values depending on the states of the bound electron. The Brown–Ravenhall continuum $\Sigma^{(+-)}$ has no threshold but crosses the imaginary axis at $\mathrm{Re}(z) = 0$. We can clearly see how the bound states are now no longer energetically overlapping with the continuum states and thus do not couple.

3.5 Relativistic Calculations on Atoms and Molecules

Differences between the results of non-relativistic and relativistic quantum chemistry are generally denoted as relativistic effects. Pyykkö has published a series of reviews [52, 184, 185] discussing relativistic effects in chemistry in great detail. Here, we limit ourselves to the simplest cases, namely, one- and two-electron atoms and the dihydrogen molecule.

3.5.1 One- and Two-Electron Atoms

Analytical solutions of one-electron atoms are known [186–188]. After a shift by the rest energy $-c^2$ (all in Hartree atomic units) to match the non-relativistic energy scale, the associated energies of the ground state is

$$E_{H,R}(Z) = c^2 \left[\sqrt{(1 - (Z/c)^2} - 1 \right], \tag{3.44}$$

where Z is the charge number of the nucleus considered to be a point charge.

For two-electron atoms, no explicit solutions exist. However, such systems have been subject to extensive work employing different approximation methods (see, e.g., Refs. [67, 189–191]). Pestka and co-workers calculated the ground-state energy by means of CI-type expansions [58, 72, 182, 189, 192, 193]. The model space was factorized according to Eq. (3.16) and the ground state fulfills the relations described in Eq. (3.17). The basis functions from which the model space was constructed are Hylleraas-type functions [72] with the Brown–Ravenhall disease cured by means of CCR.

Figure 3.4 shows a graph of the relativistic and non-relativistic energies for both, one and two-electron atoms. We can clearly see that, for light nuclei, the difference between the relativistic and non-relativistic energies are marginal. For atoms with increasing Z, however, the relativistic total electronic energies are considerably lower than the corresponding non-relativistic ones.

3.5.2 Dissociation Energy of Molecular Hydrogen

The simplest molecule one can think of is dihydrogen. Its bond energy has recently been measured to unprecedented precision [195–197]. Pachucki and co-workers [198, 199] calculated the dissociation energy of molecular hydrogen and its deuterated isotopologs HD and D_2. These theoretical results were obtained by means of relativistic and quantum electrodynamical corrections to the non-relativistic energy. The corrections are defined by a Taylor-series expansion in terms of the fine-structure constant $\alpha = 1/c$ (in Hartree atomic

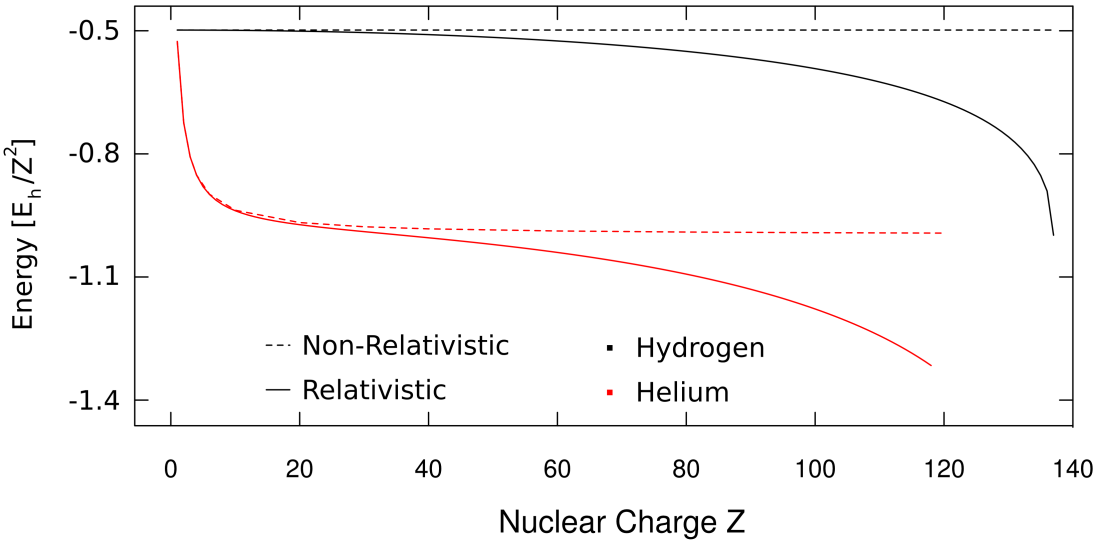

Figure 3.4: Variational electronic energies (divided by Z^2) for hydrogen- and helium-like atoms of nuclear charge numbers Z up to 137, which is the maximum possible nuclear charge number for point-like nuclei. The relativistic energies are represented by a solid line. They are analytic for the hydrogen-like atoms and taken from the work of Pestka and co-workers [72] for the helium-like atoms. The non-relativistic energies are represented by a dotted line. They are analytic for the hydrogen-like atoms and were taken from the work of Ottschofski and Kutzelnigg [194] for the helium-like atoms.

units). The agreement with experiment is remarkable, as can be seen from Table 3.1.

In their work, Pachucki and co-workers included terms of an order up to $\alpha = 4$.

3.6 Summary

In this chapter, we have attempted to provide a brief, but self-contained description of the development of the relativistic first-quantized, semi-classical many-electron theory for molecular physics and chemistry. As can be understood from the list of references, this theory is still an active field of research. In the future, we will continue to see emerging approximation methods based on one-electron spinors as well as those for highly accurate calculations based on explicitly correlated basis functions.

Table 3.1: High-accuracy theoretical and experimental data for the dissociation energy of molecular hydrogen and its deuterated isotopologs. All results reviewed are given in cm^{-1}. α denotes the fine-structure constant.

		H_2	D_2	HD
α^0	Born–Oppenheimer	36 112.5927(1)	36 746.1623(1)	36 401.9332(1)
	adiabatic	5.7711(1)	2.7725(1)	4.2509(1)
	nonadiabatic	0.4339(2)	0.1563(2)	0.3267(2)
	total α^0	36 118.7978(2)	36 749.0910(2)	36 406.5108(2)
α^2	mass-velocity	4.4273(2)	4.5125(2)	
	one-electron Darwin	-4.9082(2)	-4.9873(2)	
	two-electron Darwin	-0.5932(1)	-0.5993(1)	
	Breit	0.5422(1)	0.5465(1)	
	total α^2	-0.5319(3)	-0.5276(3)	-0.5299(4)
α^3	one-electron Lamb shift	-0.2241(1)	-0.2278(1)	
	two-electron Lamb shift	0.0166(1)	0.0167(1)	
	Araki–Sucher	0.0127(1)	0.0128(1)	
	total α^3	-0.1948(2)	-0.1983(2)	-0.1964(2)
α^4	one-loop term	-0.0016(8)	-0.0016(8)	-0.0016(8)
total		36 118.0695(10)	36 748.3633(9)	36 405.7828(10)
Ref.		[198]	[198]	[199]
exp.		36 118.0696(4)	36 748.36286(68)	36 405.78366(36)
Ref.		[195]	[196]	[197]

4

Translationally Invariant Integrals

Expressions for the calculation of intrinsic properties of molecules should be free of contributions from their overall translation. In the commonly used BO approximation, the nuclei are fixed, and thus the translational contribution is automatically separated. In several combined electron-nuclear orbital approaches [200–202] the translational dependence is eliminated automatically by fixing one or a few heavy particles. Here, we consider molecules as many-particle quantum systems with electrons and nuclei both treated as quantum particles on equal footing in the pre-BO quantum theory.

Traditionally, in rovibrational calculations, in which all nuclei are treated as quantum particles on a potential-energy surface [203–207], the first step is the separation of the Cartesian coordinates of the center of mass followed by the definition of a body-fixed frame, orientational angles, and internal coordinates. This approach results in the replacement of the original LFCC with curvilinear coordinates and the corresponding very complicated, translationally invariant rotational-vibrational Hamiltonians, see for example [208].

Less complicated translationally invariant Hamiltonians are used in full pre-BO calculations, where the original LFCC are replaced by some set of TICCs and the CMCC are separated [3, 29, 30, 34, 82, 83]. Although the resulting TICCs are rectilinear coordinates, the corresponding Hamiltonian is still complicated. It is therefore reasonable to ask whether we can make our calculations even simpler, using the original LFCC formalism without having to rely on any coordinate transformation at all. In this chapter we therefore explore the usage of LFCCs in pre-BO calculations.

If LFCCs are used one has to make sure that the energy of the overall translation of the system is eliminated. The most straightforward way is the subtraction of the kinetic-energy operator of the center of mass from the total Hamiltonian, which, however, requires the evaluation of an additional integral with mixed coordinate second derivatives [209, 210]. To avoid this additional integral evaluation we develop here an alternative approach.

In short, our computational strategy in the LFCC formalism to obtain eigenstates with various angular momentum quantum number is as follows. In our variational procedure we use basis functions, which are eigenfunctions of the total spatial angular momentum operators, $\hat{L}^2$ and $\hat{L}_z$ [30]. This is the simplest way to make sure that we obtain angular momentum eigenstates, since rotational "contamination" cannot be removed by a simple subtraction of a term from the full Hamiltonian [210,211]. Then, we investigate the effect of the parametrisation of the basis functions on the translational contamination of the total energy and correct for it during the evaluation of the integrals in the LFCC formalism. This chapter was published in Ref. [81].

4.1 Non-Relativistic Kinetic Energy

In spite of the parametrisation difficulties described in Section 2.2, we intend to use the LFCC formalism to construct the matrix representation for the quantum Hamiltonian because of its original simplicity. We repeat here only the necessary expressions from Ref. [30] and for the original integral derivation see [2].

The matrix element of the kinetic-energy operator for the Ith and Jth quasi-normalized basis (normalization with respect to the spatial basis functions) functions is [30]:

$$T_{IJ} = -\frac{1}{2}\sum_{i=1}^{N}\frac{\langle\psi_I|\,\boldsymbol{\nabla}_{\boldsymbol{r}_i}^{\mathrm{T}}\left(\boldsymbol{M}\otimes\mathbf{1}_3\right)\boldsymbol{\nabla}_{\boldsymbol{r}_i}\,|\psi_J\rangle}{|\psi_I|\,|\psi_J|} = D^{3/4}\left(\frac{p_{\boldsymbol{u}_I,\boldsymbol{u}_I}}{q_{\boldsymbol{u}_I}}\right)^{K_I}\left(\frac{p_{\boldsymbol{u}_J,\boldsymbol{u}_J}}{q_{\boldsymbol{u}_J}}\right)^{K_J}$$
$$\times\left(\frac{p_{\boldsymbol{u}_I,\boldsymbol{u}_J}}{\sqrt{q_{\boldsymbol{u}_I}q_{\boldsymbol{u}_J}}}\right)^{L}\sum_{m=0}^{\min(K_I,K_J)}\left(\frac{p_{\boldsymbol{u}_I,\boldsymbol{u}_J}p_{\boldsymbol{u}_I,\boldsymbol{u}_J}}{p_{\boldsymbol{u}_I,\boldsymbol{u}_I}p_{\boldsymbol{u}_J,\boldsymbol{u}_J}}\right)^{m}\Bigg[R_{IJ}+(K_I-m)\frac{P_{\boldsymbol{u}_I,\boldsymbol{u}_I}}{p_{\boldsymbol{u}_I,\boldsymbol{u}_I}}$$
$$+(K_J-m)\frac{P_{\boldsymbol{u}_J,\boldsymbol{u}_J}}{p_{\boldsymbol{u}_J,\boldsymbol{u}_J}}+(L+2m)\frac{P_{\boldsymbol{u}_I,\boldsymbol{u}_J}}{p_{\boldsymbol{u}_I,\boldsymbol{u}_J}}\Bigg]H_{LK_IK_Jm} \tag{4.1}$$

where $\boldsymbol{M}$ is a diagonal matrix with $M_{ii}=1/m_i$. The $H_{LK_IK_Jm}$ terms are precalculated factors

$$H_{LK_IK_Jm} = \frac{4^m(L+m+1)!}{(K_I-m)!(K_J-m)!m!(2L+2m+1)!}\times\frac{1}{\sqrt{F_{K_IL}F_{K_JL}}} \tag{4.2}$$

in order to increase efficiency and ensure numerical stability with the terms

$$F_{KL} = \sum_{m=0}^{K} \frac{4^m (L+m+1)!}{(K-m)!(K-m)!m!(2L+2m+2)!} \tag{4.3}$$

stemming from the quasi-normalization

$$|\psi_Z| = (\langle \psi_Z \, | \psi_Z \rangle)^{\frac{1}{2}} \quad \text{with} \quad Z \in \{I, J\}. \tag{4.4}$$

For more information on quasi-normalized basis functions the reader is refered to Ref. [30].

Furthermore we introduced short-hand notations for terms which we have to study in terms of their dependence on c_A as defined in Eq. (2.42). One term we have to study is

$$D = \frac{\det(2\boldsymbol{A}_I)\det(2\boldsymbol{A}_J)}{\det(\boldsymbol{A}_{IJ})\det(\boldsymbol{A}_{IJ})} \tag{4.5}$$

stemming from the origin-centered Gaussian functions, where $\boldsymbol{A}_{IJ} = \boldsymbol{A}_I + \boldsymbol{A}_J$. Furthermore we have to analyze the terms depending on the global vectors

$$p_{\boldsymbol{u}_X, \boldsymbol{u}_Y} = \boldsymbol{u}_X^T \boldsymbol{A}_{IJ}^{-1} \boldsymbol{u}_Y \quad \text{with} \quad X, Y \in \{I, J\} \tag{4.6}$$

$$P_{\boldsymbol{u}_I, \boldsymbol{u}_I} = -\boldsymbol{u}_I^T \boldsymbol{A}_{IJ}^{-1} \boldsymbol{A}_J \boldsymbol{M} \boldsymbol{A}_J \boldsymbol{A}_{IJ}^{-1} \boldsymbol{u}_I \tag{4.7}$$

$$P_{\boldsymbol{u}_J, \boldsymbol{u}_J} = -\boldsymbol{u}_J^T \boldsymbol{A}_{IJ}^{-1} \boldsymbol{A}_I \boldsymbol{M} \boldsymbol{A}_I \boldsymbol{A}_{IJ}^{-1} \boldsymbol{u}_J \tag{4.8}$$

$$P_{\boldsymbol{u}_I, \boldsymbol{u}_J} = \boldsymbol{u}_I^T \boldsymbol{A}_{IJ}^{-1} \boldsymbol{A}_J \boldsymbol{M} \boldsymbol{A}_I \boldsymbol{A}_{IJ}^{-1} \boldsymbol{u}_J, \tag{4.9}$$

the terms from the quasi-normalization

$$q_{\boldsymbol{u}_Z} = \frac{1}{2} \boldsymbol{u}_Z^T \boldsymbol{A}_Z^{-1} \boldsymbol{u}_Z \quad \text{with} \quad Z \in \{I, J\} \tag{4.10}$$

and the term

$$R_{IJ} = \frac{3}{2} \mathrm{Tr}\left[\boldsymbol{A}_{IJ}^{-1} \boldsymbol{A}_J \boldsymbol{M} \boldsymbol{A}_I\right] \tag{4.11}$$

which can be associated with the radial motion of the system [38].

Here, we immediately see that the singularity of $\boldsymbol{A}_I$ and $\boldsymbol{A}_J$ would cause terms in Eqs. (4.5) – (4.11) to be not defined. The singularity is introduced if the $c_A = 0$ selection is made to guarantee translation-free expressions.

Since the potential-energy terms in Eq. (2.3) depend only on the inter-particle distances, we can choose any $c_A > 0$ value and evaluate the matrix elements without any problem and the resulting potential-energy matrix elements are independent of the value of c_A. So, they are not discussed here any longer and are evaluated according to Ref. [30].

4.2 Identification and Elimination Strategy for the Translational Contamination

In Section 2.2, we noted that the ECGs take the same mathematical form, independent of whether we choose the LFCCs $\boldsymbol{r}$, or some $\boldsymbol{x}_{\mathrm{TICM}}$. This simple transformation behavior also transfers to the expressions of the integrals. Due to this property we can study the influence of c_A on the different terms in Eqs. (4.5) – (4.11), only the parameter matrices $\boldsymbol{A}_I, \boldsymbol{A}_J$ and $\boldsymbol{u}_I, \boldsymbol{u}_J$ corresponding to $\boldsymbol{r}$ have to be replaced by their according expressions in terms of $\boldsymbol{A}_I^{(x)}, \boldsymbol{A}_J^{(x)}$ and $\boldsymbol{u}_I^{(x)}, \boldsymbol{u}_J^{(x)}$ corresponding to $\boldsymbol{x}_{\mathrm{TICM}}$. The parameter matrices are related by the transformation given in Eqs. (2.37) – (2.38), and $\boldsymbol{A}_I^{(x)}, \boldsymbol{A}_J^{(x)}$ and $\boldsymbol{u}_I^{(x)}, \boldsymbol{u}_J^{(x)}$ have block structure, Eq. (2.43).

Firstly, let us consider the R_{IJ} term of the kinetic-energy matrix elements, Eq. (4.11), explicitly. We analyze the properties of R_{IJ} with respect to c_A using the TICC formalism:

$$\begin{aligned} R_{IJ} &= \frac{3}{2}\mathrm{Tr}\left[\boldsymbol{A}_{IJ}^{-1}\boldsymbol{A}_J\boldsymbol{M}\boldsymbol{A}_I\right] \\ &= \frac{3}{2}\mathrm{Tr}\left[(\boldsymbol{A}_{IJ}^{(x)})^{-1}\boldsymbol{A}_J^{(x)}\boldsymbol{U}_x\boldsymbol{M}\boldsymbol{U}_x^T\boldsymbol{A}_I^{(x)}\right] \end{aligned} \tag{4.12}$$

and also exploit the block structure of the matrices,

$$\boldsymbol{A}_z^{(x)} = \begin{bmatrix} \mathcal{A}_z^{(x)} & 0 \\ 0 & c_A \end{bmatrix} \quad \text{and} \quad \boldsymbol{U}_x\boldsymbol{M}\boldsymbol{U}_x^T = \begin{bmatrix} \boldsymbol{\mu}^{(x)} & 0 \\ 0 & c_M \end{bmatrix} , \tag{4.13}$$

with $z \in \{I, J, IJ\}$. The block structure of $\boldsymbol{U}_x\boldsymbol{M}\boldsymbol{U}_x^T$ follows from the conditions of translation invariance, Eqs. (2.7) and (2.8), for $\boldsymbol{x}$. We can thus separate the c_A-dependent terms in R_{IJ}:

$$R_{IJ} = R_{IJ}^{\mathrm{Int}} + \frac{3}{4}c_A c_M = \frac{3}{2}\mathrm{Tr}\left[(\mathcal{A}_{IJ}^{(x)})^{-1}\mathcal{A}_J^{(x)}\boldsymbol{\mu}^{(x)}\mathcal{A}_I^{(x)}\right] + \frac{3}{4}c_A c_M \ . \tag{4.14}$$

The c_M factor of the linear contribution is

$$c_M = (\boldsymbol{U}_x\boldsymbol{M}\boldsymbol{U}_x^T)_{N,N} = \sum_{i=1}^{N}(\boldsymbol{U}_x)_{N,i}^2/m_i = 1/m_{\mathrm{tot}} \ . \tag{4.15}$$

and thus with $c_M = 1/m_{\mathrm{tot}}$

$$R_{IJ} = R_{IJ}^{\mathrm{Int}} + \frac{3}{4}\frac{c_A}{m_{\mathrm{tot}}} \ . \tag{4.16}$$

Next, we investigate the c_A dependence of the remaining terms and factors, Eqs. (4.5) – (4.10), of the matrix representation of the kinetic energy in

the LFCC formalism, Eq. (4.1). In Eq. (4.5) we find that any contribution of c_A cancels. In Eqs. (4.10) and (4.6) we see that c_A only contributes if $c_u > 0$. Since we require $c_u = 0$, this contribution is eliminated. Finally, in Eqs. (4.7) – (4.9) we find that the contribution of c_A to the exponent matrices cancel. In short, only the R_{IJ} term, Eq. (4.11), has a non-vanishing (linear) c_A dependence in the kinetic-energy matrix element, T_{IJ}.

Thus, if $c_A = 0$ was chosen, the translational dependence vanishes, but the exponents matrices, $\boldsymbol{A}_I, \boldsymbol{A}_J$, are singular without having an inverse. Thus, for an implementation in a computer program we can choose a non-zero value for c_A, and eliminate the translational contribution explicitly by subtracting $3c_A/(4m_{\rm tot})$ from the R_{IJ} matrix element. This is a simple computational strategy which we are going to follow in the LFCC formalism.

We also note here that the direct variational optimisation of all parameters, including c_A here, would be another option that has been suggested already in the literature, e.g., in Ref. [201]. From the theoretical details presented so far, we understand that the total energy with $c_A > 0$ is always an upper bound to the total energy free of the overall translation of the system, $E_{\rm tot}(c_A) \geq E_{\rm TI}$. But as we have shown for $c_A = 0$ several parameter matrices incorporated in $E_{\rm tot}(c_A)$ are singular, so in spite of the fact that the limit exists, the application in a computer program is problematic ($c_u = 0$ is chosen throughout the discussion).

This explains our preference for the approach developed here, which releases the translation-free condition for the basis functions and corrects for the translational contamination in the kinetic energy explicitly for each basis function, $I = 1, 2, \ldots, N$. The details of our algorithm are:

1. Generate, optimize or read in the $\alpha_{I,ij}$ values for

$$i = 1, 2, \ldots, N, \quad j = i+1, i+2, \ldots, N \ .$$

2. Construct the elements of the exponent matrix in the LFCC formalism as

$$(\boldsymbol{A}_I)_{ij} = -\alpha_{I,ij}(1-\delta_{ij}) + \left(\sum_{k=1, k\neq i}^{N} \alpha_{I,ik} \right) \delta_{ij} + c_A \frac{m_i}{m_{\rm tot}} \frac{m_i}{m_{\rm tot}}$$

   with $i, j = 1, 2, \ldots, N$ and $c_A > 0$.

3. Due to the $c_A > 0$ choice the matrix $\boldsymbol{A}_I$ is non-singular, and $1/\det(\boldsymbol{A}_I)$ and $\boldsymbol{A}_I^{-1}$ can be calculated. At the same time the total kinetic energy contains some translational contamination.

4. The translational contamination is eliminated by replacing R_{IJ}, Eq. (4.11), with $R_{IJ} - 3c_A/(4m_{\mathrm{tot}})$ in the expression of the kinetic-energy matrix element, T_{IJ}, Eq. (4.1).

Throughout this computational strategy for the elimination of the translational contamination in the LFCC formalism we have $c_u = 0$, $c_A > 0$.

4.3 Numerical Examples

In this section we present numerical applications using the LFCC formalism. The appearance of the translational contamination and its elimination according to the strategy described in Section 4.2 are demonstrated.

Our test cases are the lowest energy levels of the para-H_2 ($L = 0, p = +1$) and the ortho-H_2 ($L = 1, p = -1$) molecules both in the singlet electronic state. These are the two lowest-energy rotational states of the hydrogen molecule. L is the total spatial (orbital and rotational) quantum number and p is the parity. The wave functions are obtained by a direct solution of the linear variational problem using 1500 basis functions with an optimized parametrisation taken from [30]. Here we use the LFCC formalism exclusively, thus all parameters were transformed first to the LFCC representation, according to Eqs. (2.37) and (2.38). The exponent matrix for each basis function was constructed according to Eq. (2.41). During this transformation we were free to choose c_A to simulate different levels of translational contamination, while $c_u = 0$ was fixed. We used the same c_A value for each basis function. Table 4.1 collects the results of our numerical calculations.

The first column of Table 4.1 lists the five values of c_A which have been studied. The next two columns provide the total energies of the system with and without translational contamination, respectively. The last two columns list the kinetic-energy contribution with and without translational contamination, respectively. We observe that those energies, which contain translational contributions (columns 2 and 4) depend on the value of c_A and that an increase of the value of c_A causes an increase of the energy, according to Section 4.2. We also note that the corrected energies (columns 3 and 5) are independent of c_A. Furthermore, we list the translational correction given in Eq. (4.16). Hence, these results give a numerical confirmation that we have identified and eliminated the translational contamination depending on the c_A basis function parameter, while choosing $c_u = 0$.

Table 4.1: The appearance and elimination of the translational contamination of the total pre-Born–Oppenheimer energy using the laboratory-fixed-Cartesian-coordinate (LFCC) formalism. The ground-state energies, in E_h, of the singlet, para-H_2 ($L = 0, p = +1$) and ortho-H_2 ($L = 1, p = -1$) molecules are calculated using the basis function parameter set of Ref. [30]. The corresponding translationally invariant energies are $E_{\mathrm{TI}}(L = 0) = -1.164\,025\,026$ E_h and $E_{\mathrm{TI}}(L = 1) = -1.163\,485\,167$ E_h, respectively.[a,d] Results corrected for the translational contamination of the LFCC formalism are indicated with the "corr." subscript.

c_A^b	E_{LF}^c	$E_{\mathrm{LF,corr.}}^c$	$\langle\Psi_{\mathrm{LF}}\vert\hat{T}_{\mathrm{LF}}\vert\Psi_{\mathrm{LF}}\rangle^c$	$\langle\Psi_{\mathrm{LF}}\vert\hat{T}_{\mathrm{LF}}\vert\Psi_{\mathrm{LF}}\rangle_{\mathrm{corr.}}^c$	$\delta_{\mathrm{Tr.}}^c$
para-H_2 $L = 0^d$					
0.01	-1.164022985	-1.164025026	1.164027045	1.164025004	0.000002041
0.10	-1.164004614	-1.164025026	1.164045416	1.164025004	0.000020412
0.50	-1.163922966	-1.164025026	1.164127064	1.164025004	0.000102060
1.00	-1.163820906	-1.164025026	1.164229124	1.164025004	0.000204120
2.00	-1.163616786	-1.164025026	1.164433245	1.164025004	0.000408240
ortho-H_2 $L = 1^d$					
0.01	-1.163483125	-1.163485167	1.163487203	1.163485161	0.000002041
0.10	-1.163464755	-1.163485167	1.163505573	1.163485161	0.000020412
0.50	-1.163383107	-1.163485167	1.163587222	1.163485161	0.000102060
1.00	-1.163281047	-1.163485167	1.163689282	1.163485161	0.000204120
2.00	-1.163076927	-1.163485167	1.163893402	1.163485161	0.000408240

[a] The parametrisation of the internal basis set ($\mathcal{P}$) was taken from Ref. [30]. The number of basis functions was $1\,500$, the order of the polynomial prefactors was $2K \in [0, 20]$. The virial ratios, $\eta = |1 + \langle\Psi|\hat{V}|\Psi\rangle / (2\langle\Psi|\hat{T}|\Psi\rangle)|$ corresponding to the translationally invariant energies obtained with these basis sets are $\eta(L = 0) = 1.4 \cdot 10^{-8}$ and $\eta(L = 1) = 3.2 \cdot 10^{-9}$. For the proton-electron mass ratio the value $m_p/m_e = 1\,836.152\,672\,47$ was used, and thus $m_{\mathrm{tot}} = 3\,674.305\,344\,94$ $m_e{=}2m_p + 2m_e$.

[b] Free parameter of the explicitly correlated Gaussian basis functions expressed in LFCCs. The same c_A value was used for each basis function in the basis set. The value $c_u = 0$ was used throughout the calculations.

[c] E_{LF} , Ψ_{LF} : Eigenvalue and eigenfunction of the full Hamiltonian expressed in LFCC using the basis function parameters $(\mathcal{P}, c_A)$.
$\hat{T}_{\mathrm{LF}}$: The kinetic-energy operator expressed in LFCCs.
"corr." : Correction for the translational contamination in the LFCC formalism, as explained in Section 2.1.
$\delta_{\mathrm{Tr.}} = 3c_A/(4m_{\mathrm{tot}})$: Translational contamination of the kinetic energy, see Eq. (4.16). The corrected, translation-free value is obtained as $E_{\mathrm{LF,corr.}} = E_{\mathrm{LF}} - \delta_{\mathrm{Tr.}}$.

[d] For comparison, the best theoretical values available energy values in the literature are $E(L = 0) = -1.164\,025\,030$ [E_h] and $E(L = 1) = -1.163\,485\,172$ [E_h] [191].

4.4 Summary

Instead of transforming the coordinates we accounted for the translational and rotational invariances of the isolated many-particle problem by using an appropriate form and parametrisation of the basis functions in the variational procedure. The basis functions were constructed using ECGs and the GVR, and as an extension of our earlier work [30] we focused here on the usage of LFCCs and the problem of translational invariance.

First of all, we observed that it is impossible to parametrize explicitly correlated Gaussian functions (ECGs) in such a way that the total system is at rest in LFCCs and at the same time the basis functions are square-integrable with a non-vanishing norm.

Fortunately, it was possible to devise a simple computational strategy to circumvent this problem. So, for the sake of the stability of the numerical computations we released the translational constraint on the basis functions, by choosing a non-zero value for the free parameter, c_A, in the LFCC parametrisation of the ECG exponents, and then explicitly corrected for the translational contamination in the kinetic-energy integral expressions. This correction term is a simple constant, which depends linearly on c_A. Its form was derived by considering a few mathematical relationship of the formalism: a) the properties of the linear transformation between LFCCs and translationally invariant and center-of-mass Cartesian coordinates (TICMCCs); b) the corresponding transformation of the basis function parameter matrices; c) the fact that the parameter matrices are block diagonal in TICMCCs.

It was also shown that the uncorrected total energy $E_{\mathrm{tot}}(c_A)$ is an upper bound to the translation-free total (intrinsic) energy. Thus, in principle, we could obtain this value by the variational optimisation of the LFCC parametrisation (implicitly including the c_A value in the optimisation). We prefer, however, our explicit treatment for the elimination of the translational contamination, because in the $c_A = 0$ limit the exponent matrix, $\boldsymbol{A}$, of each ECG would be singular, and thus the numerical evaluation of $1/\det(\boldsymbol{A})$ and $\boldsymbol{A}^{-1}$ would be impossible.

Finally, to demonstrate the numerical applicability of our approach we calculated the lowest two rotational levels of the singlet hydrogen molecule corresponding to the para and ortho proton spin states, respectively.

The presented LFCC formalism with the explicit translational contamination correction is an alternative but equivalent to the traditional approaches using some set of TICC with the Cartesian coordinates of the center of mass explicitly

separated already in the Hamiltonian, e.g. [30, 34]. The simplicity of the LFCCs is an appealing choice for the variational solution of the Schrödinger equation with the non-relativistic Hamiltonian. Furthermore, one can think of more complicated operators for which the usage of the simplest possible coordinate representation and the avoidance of any coordinate transformation is more than just a comfortable option.

5

Transition Dipole Moments using Translationally Invariant Integrals

In this chapter we are interested in the evaluation of the electric transition dipole moment. In a semi-classical picture, it mediates transitions between two rovibrational or rovibronic states of a molecule induced by the electric component of electromagnetic-radiation. Nowadays, most calculations rely on the BO approximation which separates the electronic and nuclear degrees of freedom. For small systems, however, very accurate calculations can be carried out if the BO approximation is avoided. Simultaneous description of electrons and nuclei using ECGs has been pioneered by the Adamowicz group [3,31] and Suzuki and Varga [2,35,36]. In the present work we follow these lines and extend our earlier work [29,30,81]. The basis functions are constructed using ECGs [79, 80, 212–214] and the GVR [2, 35, 36] in order to ensure that the wave function is an eigenfunction of the total spatial angular momentum operators, $\hat{L}^2$ and $\hat{L}_z$, and parity. The basis function parameters are optimized variationally through stochastic sampling. Rather than relying on a set of Cartesian coordinates which separates the TICC from the center of mass, we use LFCC. Any translational contamination to the total energy is eliminated from the integrals as presented in chapter 4 [81]. We illustrate in this chapter that this scheme can be applied to the calculation of molecular properties such as the electric transition dipole moment. Previous calculations in the literature of the electric transition dipole moment in a pre-BO framework have been performed in TICC [215, 216]. This chapter was published in Ref. [217].

5.1 Electric Permanent and Transition Dipole Moments in Pre-BO Theory

In this chapter, we are interested in the components of the electric transition dipole moment in pre-BO theory. In systems containing N particles the electric dipole moment operator is defined as

$$\hat{\boldsymbol{\mu}} = \sum_{i=1}^{N} q_i \boldsymbol{r}_i \tag{5.1}$$

where q_i and $\boldsymbol{r}_i$ are the electric charge and position of particle i, respectively. This operator has odd parity and describes both permanent and transition dipole moments. But although being closely related, there are significant conceptual differences between the two types of dipole moments which we will discuss in this section.

Molecules described in the BO approximation have generally a non-zero permanent electric dipole moment (i.e. they are polar). The only exceptions are molecules which are non-polar due to certain symmetry properties contained within the molecular structure (e.g. methane, benzene or H_2). This illustrates how closely the molecular structure and the permanent electric dipole are related. The concept of structure (or shape) where the nuclei form a rigid scaffold, which is stabilized by the electrons, is the core concept in chemistry and clearly defined in the BO picture. Yet, there is currently no complete understanding how this concept is to be interpreted in a pre-BO framework since the nuclei are not fixed but treated as quantum particles to which particle densities are assigned. Several authors have been discussing the subject in great detail [82, 83, 200, 202, 218–231].

Generally, the symmetry properties of a pre-BO wave function are enough to gain some information about pre-BO permanent dipole moments. The total pre-BO wave function is an eigenfunction of the parity operator if no external potential is present. This is due to the isotropy of space and the resulting conservation of the total spatial angular momentum. The parity of the pre-BO wave function together with the odd parity of the dipole moment operator results in an integral over a function with ungerade symmetry and therefore always evaluates to zero. Yet, the squared length of the dipole moment

$$\hat{\boldsymbol{\mu}}^{\mathrm{T}} \hat{\boldsymbol{\mu}} = \sum_{j=1}^{N} \sum_{i=1}^{N} q_i q_j \boldsymbol{r}_i^{\mathrm{T}} \boldsymbol{r}_j \ , \tag{5.2}$$

has even parity. Thus, the integral of the squared length has gerade symmetry and is not strictly zero. We can conclude from this that a pre-BO wave function

has no permanent dipole moment but can still be polarized. This indicates that the permanent dipole does not exist due to the symmetry properties of the wave function. This idea might be investigated in later work. Another method to determine the permanent transition dipole moment (and higher-order electric properties) was presented by Cafiero, Bubin and Adamowicz [34]. They included the interaction energy of an external electric field with the permanent electric dipole in the total Hamiltonian and extrapolated the zero field energy and electric properties from results at various field strengths.

In contrast to the status of the permanent dipole moment, which is related to the classical chemical structure concept, the evaluation of the electric transition dipole moment, which is a spectroscopic quantity, is more straightforward in the pre-BO theory. The pre-BO wave function already contains not only the electronic but also the rotational-vibrational degrees of freedom.

The subscripts i and f denote the initial and final states of a transition. Transitions mediated through an electric transition dipole feature a set of selection rules such that

$$\boldsymbol{\mu}_{\mathrm{if}} = \langle \Psi_{\mathrm{i}} | \, \hat{\boldsymbol{\mu}} \, | \Psi_{\mathrm{f}} \rangle \neq 0 \ . \tag{5.3}$$

Electric transition dipole moments are complex three-vectors of the form $\boldsymbol{\mu} = (\mu_x, \mu_y, \mu_z)^{\mathrm{T}}$. The length of the electric transition dipole moment is then

$$|\boldsymbol{\mu}| = (\boldsymbol{\mu}^{\dagger} \cdot \boldsymbol{\mu})^{1/2} \ . \tag{5.4}$$

For the spatial angular quantum numbers $(L_{\mathrm{i}}, M_{L\mathrm{i}})$ and $(L_{\mathrm{f}}, M_{L\mathrm{f}})$, with natural parity $p_{\mathrm{i}} = (-1)^{L_{\mathrm{i}}}$ and $p_{\mathrm{f}} = (-1)^{L_{\mathrm{f}}}$, the integral in Eq. (5.3) is non-vanishing if $L_{\mathrm{i}} - L_{\mathrm{f}} \in \{+1, -1\}$ and for $M_{L\mathrm{i}} - M_{L\mathrm{f}} \in \{+1, 0, -1\}$. The selection rules for M_L are related to different polarizations of the absorbed/emitted light. For the spin quantum numbers one finds $S_{\mathrm{i}} - S_{\mathrm{f}} = M_{S\mathrm{i}} - M_{S\mathrm{f}} = 0$. Transitions which do not fulfill these selection rules are either symmetry forbidden (selection rules related to L and M_L) or spin forbidden (selection rules related to S and M_S). Spin forbidden transitions become allowed if different spin states mix, e.g, if relativistic effects are considered. Symmetry forbidden transitions become allowed with respect to transitions mediated by means of higher-order electric transition multi-poles.

Furthermore, degenerate substates have to be considered: Transitions involve all rotational substates Ψ_{L,M_L} with $-L \leq M_L \leq L$ and all possible transitions for which Eq. (5.3) is fulfilled. The squared length of the transition dipole

moment is then obtained as [232]

$$|\langle \Psi_{L_\mathrm{i}} \, |\hat{\boldsymbol{\mu}}| \, \Psi_{L_\mathrm{f}} \rangle|^2 = \sum_{j=-L_\mathrm{i}}^{L_\mathrm{i}} \sum_{k=-L_\mathrm{f}}^{L_\mathrm{f}} |\langle \Psi_{L_\mathrm{i},j} \, |\hat{\boldsymbol{\mu}}| \, \Psi_{L_\mathrm{f},k} \rangle|^2 \quad . \tag{5.5}$$

5.2 Evaluation of the Electric Transition Dipole Moment Integrals

In this section we focus on the electric transition dipole moment and its determination using ECGs with the GVR in an LFCC pre-BO framework. We start by presenting two different forms of the electric transition dipole moment, commonly known as the velocity and the length representation. The two representations are equivalent only for the exact wave function. Then, expressions are presented for the transition dipole integrals.

5.2.1 Velocity and Length Representation

The transition dipole moment between the "i" initial and "f" final state is

$$\boldsymbol{\mu}_\mathrm{if}^{(\mathrm{l})} = \sum_{j=1}^{N} \langle \Psi_\mathrm{i} \, |q_j \boldsymbol{r}_j| \, \Psi_\mathrm{f} \rangle \tag{5.6}$$

in the "length" (l) representation, while it can also be evaluated using the "velocity" (v) representation [233]

$$\boldsymbol{\mu}_\mathrm{if}^{(\mathrm{v})} = -\frac{1}{(E_\mathrm{i} - E_\mathrm{f})} \sum_{j=1}^{N} \left\langle \Psi_\mathrm{i} \left| \frac{q_j}{m_j} \boldsymbol{\nabla}_{\boldsymbol{r}_j} \right| \Psi_\mathrm{f} \right\rangle \quad . \tag{5.7}$$

The equivalence of Eqs. (5.6) and (5.7) can be shown with the help of the commutation relation

$$[\hat{H}, q_j \boldsymbol{r}_j] = -\frac{q_j}{m_j} \boldsymbol{\nabla}_{\boldsymbol{r}_j} \quad . \tag{5.8}$$

Acting with the initial state from the left and the final state from the right leads to the integral

$$\langle \Psi_\mathrm{i}| \, [\hat{H}, q_j \boldsymbol{r}_j] \, |\Psi_\mathrm{f}\rangle = - \langle \Psi_\mathrm{i}| \, \frac{q_j}{m_j} \boldsymbol{\nabla}_{\boldsymbol{r}_j} \, |\Psi_\mathrm{f}\rangle \quad . \tag{5.9}$$

Exploiting the fact, that the inital and the final state are eigenfunction of the Hamiltonian we find

$$(E_\mathrm{i} - E_\mathrm{f}) \, \langle \Psi_\mathrm{i}| \, q_j \boldsymbol{r}_j \, |\Psi_\mathrm{f}\rangle = - \langle \Psi_\mathrm{i}| \, \frac{q_j}{m_j} \boldsymbol{\nabla}_{\boldsymbol{r}_j} \, |\Psi_\mathrm{f}\rangle \tag{5.10}$$

where $\mathrm{E_i}$ and $\mathrm{E_f}$ are the energies of the initial and the final states, respectively. Eq. (5.10) can be easily rearranged into

$$\langle\Psi_\mathrm{i}|\, q_j \boldsymbol{r}_j \,|\Psi_\mathrm{f}\rangle = -\frac{1}{(\mathrm{E_i}-\mathrm{E_f})}\,\langle\Psi_\mathrm{i}|\,\frac{q_j}{m_j}\boldsymbol{\nabla}_{\boldsymbol{r}_j}\,|\Psi_\mathrm{f}\rangle \tag{5.11}$$

which are the individual terms of the sums in Eqs. (5.6) and (5.7). The equality of the two sides in Eq. (5.11) holds only for the exact wave functions. At the same time, no conclusion can be made [234], which representation is less sensitive to approximations in the wave functions.

5.2.2 Evaluation of the Integrals

The following three-step evaluation procedure used already for the evaluation of the overlap, kinetic and Coulomb potential-energy integrals [30] is employed in the present work for the calculation of the transition dipole moment integrals in both the length and velocity representation. The three steps for some operator $\hat{O}$ are:

1. Evaluate the integrals of $\hat{O}$ with the generating functions of Eq. (2.35):

$$I_1 = \langle g(\boldsymbol{r}, \boldsymbol{A}_I, a_I \boldsymbol{u}_I \otimes \boldsymbol{\varepsilon}_I)|\,\hat{O}\,|g(\boldsymbol{r}, \boldsymbol{A}_J, a_J \boldsymbol{u}_J \otimes \boldsymbol{\varepsilon}_J)\rangle \quad . \tag{5.12}$$

2. Evaluate the derivatives at $a_I = a_J = 0$:

$$I_2 = \frac{\partial^{2K_I+L_I}}{\partial a_I^{2K_I+L_I}}\frac{\partial^{2K_J+L_J}}{\partial a_J^{2K_J+L_J}} I_1(a_I, a_J)\bigg|_{a_I=a_J=0} \quad . \tag{5.13}$$

3. Evaluate the angular integrals:

$$I_3 = \frac{1}{B_{K_I L_I} B_{K_J L_J}} \int d\hat{\boldsymbol{\varepsilon}}_I \int d\hat{\boldsymbol{\varepsilon}}_J Y^{L_I *}_{M_{LI}}(\hat{\boldsymbol{\varepsilon}}_I) Y^{L_J}_{M_{LJ}}(\hat{\boldsymbol{\varepsilon}}_J) I_2(\boldsymbol{\varepsilon}_I, \boldsymbol{\varepsilon}_J) \quad . \tag{5.14}$$

In order to avoid numerical instabilities, quasi-normalized basis functions are used, i.e., each basis function is normalized with respect to its spatial part. The (I^{th},J^{th}) matrix elements for operator $\hat{O}$ are then

$$[O]_{IJ} = \frac{\left\langle \phi_I^{(L_I,M_{LI})}(\boldsymbol{r}; \boldsymbol{A}_I, \boldsymbol{u}_I, K_I)\right|\,\hat{O}\,\left|\phi_J^{(L_J,M_{LJ})}(\boldsymbol{r}; \boldsymbol{A}_J, \boldsymbol{u}_J, K_J)\right\rangle}{\left|\phi_I^{(L_I,M_{LI})}\right|\,\left|\phi_J^{(L_J,M_{LJ})}\right|} \quad , \tag{5.15}$$

where $\left|\phi_I^{(L_I,M_{LI})}\right|$ and $\left|\phi_J^{(L_J,M_{LJ})}\right|$ are the normalization factors and the square brackets denote the matrix representation of $\hat{O}$.

In this chapter the integrals in Eqs. (5.6) and (5.7) are evaluated with ECG basis functions in the GVR. Instead of the original Cartesian components $\alpha \in \{x, y, z\}$ we use transformed components

$$(\hat{\mathbf{\Omega}}_j)_+ = (\hat{\mathbf{\Omega}}_j)_x - i(\hat{\mathbf{\Omega}}_j)_y \tag{5.16}$$

$$(\hat{\mathbf{\Omega}}_j)_- = (\hat{\mathbf{\Omega}}_j)_x + i(\hat{\mathbf{\Omega}}_j)_y \tag{5.17}$$

$$(\hat{\mathbf{\Omega}}_j)_z = (\hat{\mathbf{\Omega}}_j)_z \tag{5.18}$$

collected under the label $\beta \in \{+, -, z\}$ where $\hat{\mathbf{\Omega}}_j \in \{\hat{\boldsymbol{\mu}}^{(\mathrm{v})}, \hat{\boldsymbol{\mu}}^{(\mathrm{l})}\}$. The transformed components are especially convenient for evaluating the angular integrals. For the complete derivation we may refer the reader to the supporting information of Ref. [217].

Following this integration scheme we get the expression:

$$\begin{aligned}
[\mathbf{\Omega}_\beta]_{IJ} &= \frac{\left\langle \phi_I^{(L_I,M_{LI})} \middle| \hat{\mathbf{\Omega}}_\beta \middle| \phi_J^{(L_J,M_{LJ})} \right\rangle}{\left| \phi_I^{(L_I,M_{LI})} \right| \left| \phi_J^{(L_J,M_{LJ})} \right|} \\
&= \left(\frac{\det(2\boldsymbol{A}_I)\det(2\boldsymbol{A}_J)}{\det(\boldsymbol{A}_{IJ})\det(\boldsymbol{A}_{IJ})} \right)^{3/4} \left(\frac{p_{II}}{q_I} \right)^{K_I} \left(\frac{p_{JJ}}{q_J} \right)^{K_J} \left(\frac{p_{IJ}}{\sqrt{q_I q_J}} \right)^{L_J} (q_I)^{-1/2} \\
&\times C_1^\beta(L_J, L_I, M_{LJ}, M_{LI}) \left[G_I^{\mathbf{\Omega}} \sum_{m=0}^{\min(K_I,K_J)} \left(\frac{p_{IJ}p_{IJ}}{p_{II}p_{JJ}} \right)^m H_1(m, K_I, K_J, L_J) \right. \\
&\left. + \frac{G_J^{\mathbf{\Omega}} p_{II}}{p_{IJ}} \sum_{m=1}^{\min(K_I+1,K_J)} \left(\frac{p_{IJ}p_{IJ}}{p_{II}p_{JJ}} \right)^m H_2(m, K_I, K_J, L_I) \right] \\
&+ \left(\frac{\det(2\boldsymbol{A}_I)\det(2\boldsymbol{A}_J)}{\det(\boldsymbol{A}_{IJ})\det(\boldsymbol{A}_{IJ})} \right)^{3/4} \left(\frac{p_{II}}{q_I} \right)^{K_I} \left(\frac{p_{JJ}}{q_J} \right)^{K_J} \left(\frac{p_{IJ}}{\sqrt{q_I q_J}} \right)^{L_I} (q_J)^{-1/2} \\
&\times C_2^\beta(L_I, L_J, M_{LI}, M_{LJ}) \left[\frac{G_I^{\mathbf{\Omega}} p_{JJ}}{p_{IJ}} \sum_{m=1}^{\min(K_I,K_J+1)} \left(\frac{p_{IJ}p_{IJ}}{p_{II}p_{JJ}} \right)^m H_2(m, K_J, K_I, L_J) \right. \\
&\left. + G_J^{\mathbf{\Omega}} \sum_{m=0}^{\min(K_I,K_J)} \left(\frac{p_{IJ}p_{IJ}}{p_{II}p_{JJ}} \right)^m H_1(m, K_I, K_J, L_I) \right]
\end{aligned} \tag{5.19}$$

where $\beta \in \{+, -, z\}$, $\boldsymbol{A}_{IJ} = \boldsymbol{A}_I + \boldsymbol{A}_J$ and the terms related to the overlap of the ECGs are

$$p_{XY} = \frac{1}{2} \boldsymbol{u}_X \boldsymbol{A}_{IJ}^{-1} \boldsymbol{u}_Y \quad \text{with} \quad X, Y \in \{I, J\} \ . \tag{5.20}$$

The length and velocity dependence is contained within the factors

$$G_I^{\mathbf{\Omega}} = \begin{cases} \sum_n \dfrac{q_n(\boldsymbol{A}_J \boldsymbol{A}_{IJ}^{-1} \boldsymbol{v}_I)_n}{m_n(E_\mathrm{i} - E_\mathrm{f})} & \text{if } \mathbf{\Omega} = \boldsymbol{\mu}_\mathrm{if}^{(\mathrm{v})} \\ \sum_n q_n(\boldsymbol{A}_{IJ}^{-1} \boldsymbol{v}_I)_n & \text{if } \mathbf{\Omega} = \boldsymbol{\mu}_\mathrm{if}^{(\mathrm{l})} \end{cases} \tag{5.21}$$

$$G_J^{\mathbf{\Omega}} = \begin{cases} \sum_n \dfrac{q_n(\boldsymbol{A}_J \boldsymbol{A}_{IJ}^{-1} \boldsymbol{v}_J - \boldsymbol{v}_J)_n}{m_n(E_\mathrm{i} - E_\mathrm{f})} & \text{if } \mathbf{\Omega} = \boldsymbol{\mu}_\mathrm{if}^{(\mathrm{v})} \\ \sum_n q_n(\boldsymbol{A}_{IJ}^{-1} \boldsymbol{v}_J)_n & \text{if } \mathbf{\Omega} = \boldsymbol{\mu}_\mathrm{if}^{(\mathrm{l})} \end{cases} \tag{5.22}$$

while the different contributions from the "x", "y" and "z" components are contained within the factors

$$\begin{aligned} &C_1^\beta(L_J, L_I, M_{LJ}, M_{LI}) \\ &= \begin{cases} \left(\frac{(L_J+M_{LJ}+1)(L_J-M_{LJ}+1)}{(2L_J+1)(2L_J+3)}\right)^{1/2} \delta_{M_{LJ},M_{LI}} \delta_{L_I,L_J+1} & \text{if } \beta = z \\ \left(\frac{(L_J-M_{LJ}+2)(L_J-M_{LJ}+1)}{(2L_J+1)(2L_J+3)}\right)^{1/2} \delta_{M_{LJ},M_{LI}+1} \delta_{L_I,L_J+1} & \text{if } \beta = + \\ -\left(\frac{(L_J+M_{LJ}+2)(L_J+M_{LJ}+1)}{(2L_J+1)(2L_J+3)}\right)^{1/2} \delta_{M_{LI},M_{LJ}+1} \delta_{L_I,L_J+1} & \text{if } \beta = - \end{cases} \end{aligned} \tag{5.23}$$

$$\begin{aligned} &C_2^\beta(L_I, L_J, M_L, M_{LJ}) \\ &= \begin{cases} \left(\frac{(L_I+M_{LI}+1)(L_I-M_{LI}+1)}{(2L_I+1)(2L_I+3)}\right)^{1/2} \delta_{M_{LI},M_{LJ}} \delta_{L_J,L_I+1} & \text{if } \beta = z \\ -\left(\frac{(L_I+M_{LI}+2)(L_I+M_{LI}+1)}{(2L_I+1)(2L_I+3)}\right)^{1/2} \delta_{M_{LJ},M_{LI}+1} \delta_{L_J,L_I+1} & \text{if } \beta = + \\ \left(\frac{(L_I-M_{LI}+2)(L_I-M_{LI}+1)}{(2L_I+1)(2L_I+3)}\right)^{1/2} \delta_{M_{LI},M_{LJ}+1} \delta_{L_J,L_I+1} & \text{if } \beta = - \end{cases} \end{aligned} \tag{5.24}$$

which include the selection rules for the different transitions. Note that the selection rules emerge naturally from the derivation and are not included in a technical fashion. In order to increase the efficiency of the calculations the following terms are precalculated:

$$\begin{aligned} &H_1(m, K_1, K_2, L) \\ &= \frac{B_{mL}}{(K_1-m)!(K_2-m)!(2m+L)!} \left[F(K_1, L+1) F(K_2, L)\right]^{-1/2} \end{aligned} \tag{5.25}$$

$$\begin{aligned} &H_2(m, K_1, K_2, L) \\ &= \frac{B_{(m-1)L}}{(K_1-m+1)!(K_2-m)!(2m+L-2)!} \left[F(K_1, L) F(K_2, L-1)\right]^{-1/2} \quad . \end{aligned} \tag{5.26}$$

The remaining terms

$$q_X = \frac{1}{2}\boldsymbol{u}_X^{\mathrm{T}}\boldsymbol{A}_X^{-1}\boldsymbol{u}_X \quad \text{with} \quad X \in \{I, J\} \tag{5.27}$$

$$F(K,L) = \sum_{m=0}^{K} \frac{4^m B_{mL}}{4^K (K-m)!(K-m)!(L+2m)!} \tag{5.28}$$

originate from the quasi-normalization.

Finally, we perform the back transformations from Eqs. (5.16) and (5.17) in order to obtain the original (x, y, z) Cartesian components

$$\begin{aligned} &C_1^{\beta}(L_J, L_I, M_{LJ}, M_{LI}) \\ &= \begin{cases} \frac{1}{2}[C_1^-(L_J, L_I, M_{LJ}, M_{LI}) + C_1^+(L_J, L_I, M_{LJ}, M_{LI})] & \text{if } \beta = x \\ \frac{i}{2}[C_1^-(L_J, L_I, M_{LJ}, M_{LI}) - C_1^+(L_J, L_I, M_{LJ}, M_{LI})] & \text{if } \beta = y \end{cases}, \end{aligned} \tag{5.29}$$

$$\begin{aligned} &C_2^{\beta}(L_I, L_J, M_L, M_{LJ}) \\ &= \begin{cases} \frac{1}{2}[C_2^-(L_I, L_J, M_L, M_{LJ}) + C_2^+(L_I, L_J, M_L, M_{LJ})] & \text{if } \beta = x \\ \frac{i}{2}[C_2^-(L_I, L_J, M_L, M_{LJ}) - C_2^+(L_I, L_J, M_L, M_{LJ})] & \text{if } \beta = y \end{cases}. \end{aligned} \tag{5.30}$$

This concludes the derivation for the integral expressions for the electric transition dipole moment. The squared length is then calculated according to Eq. (5.5) as the sum of the squared lengths of the allowed transitions among the degenerate substates of the initial and final state.

5.2.3 Elimination of the Translational Contamination

The contributions from c_A to Eq. (5.19) are identified by substituting $\boldsymbol{A}$ and $\boldsymbol{u}$ with their corresponding expressions in terms of $\boldsymbol{A}^{(x)}$ and $\boldsymbol{u}^{(x)}$ according to Eqs. (2.37) and (2.38). Most of the contributions of c_A cancel from the integral expressions due to the quasi-normalization of the basis functions. The remaining contributions of c_A are then eliminated from the integral expressions by subtraction if necessary. There are two terms through which such a contamination might be introduced in the dipole integrals: G_I^{Ω}, Eq. (5.21), and G_J^{Ω}, Eq. (5.22). Performing the substitutions, we find that the results of Eq. (5.19) are free from any translational contamination if c_u is zero. This allows us to evaluate and implement the transition dipole integrals using the LFCC formalism.

The advantages of performing calculations in LFCC are three-fold. First, one does not have to choose a set of TICC, which introduces some ambiguity. Furthermore the physical picture is more intuitive, and most importantly, the

many-particle integrals are evaluated more easily [27]. An advantage of a TICC pre-BO framework is that the dimension of $\boldsymbol{A}$ is $(N-1)\times(N-1)$ rather than $N\times N$ so some of the involved matrix operations become slightly computationally less expensive. TICCs are also more suited to represent correlations paths among certain particles [30].

Since the parameters c_A and c_u are independent for the specific choice of the transformation $\boldsymbol{U}_x$, LFCC-TICC hybrid methods can be imagined where specific correlations paths can be included into the parametrization. This idea might be investigated in later work.

The numerical implementation of the dipole integrals has been validated for HT^+ using the results of Bekbaev and co-workers [216] and Tian and co-workers [215] as references.

5.3 Numerical Results

In this section we present numerical results for the electric transition dipole moment calculated for the $H_2 = \{p^+, p^+, e^-, e^-\}$ molecule. Pure rotational dipole transitions of H_2 with $\Delta L = \pm 1$ are not possible because of the alternating ortho ($S_p = 1$) and para ($S_p = 0$) states in the ground electronic state [229]. There can be however non-vanishing transition moments between rovibronic levels of different electronic states with the same electron spin state (S_e). Thus, we have considered rovibronic transitions between energy levels assignable to the two lowest-lying singlet electronic states in the BO theory, $X\ ^1\Sigma_g^+$ and $B\ ^1\Sigma_u^+$ (Figure 5.1). For these transitions the electronic dipole transition function has already been calculated [235], but we are not aware of any calculation of the dipole transition moments using this dipole transition function in a non-adiabatic framework. In the present work, we do not rely on any dipole moment function, but evaluate the electric transition dipole moments by directly evaluating the transition dipole integrals with the pre-BO wave functions. The wave functions used in this chapter have been successfully applied for the calculation of resonances in a recent work [29]. The employed parameter sets corresponding to 2250 basis functions for each state are provided in the supporting information of Ref. [217]. The mass of the proton m_p was chosen in terms of the electron mass m_e as $m_p/m_e = 1836.15267247$ [236].

Table 5.1 lists the energies corresponding to the basis functions which we use for the considered rovibronic states. We use reference values from the literature to assess the quality of our parameter sets. All parameter sets are either comparable to the best available reference values ($X\ ^1\Sigma_g^+$) or better

Table 5.1: Energies of the $H_2 = \{p^+, p^+, e^-, e^-\}$ parameter sets used in this chapter together with references from the literature. The size of the parameter sets was 2250 in all cases. All results are in atomic units. Only vibrational ground states are considered.

$(L,p,S_p,S_e)^a$	E $[E_h]^b$	η $[10^{-9}]^c$	E_{Ref}-E $[\mu E_h]$	Ref.	Assignment^d
(0,+1,0,0)	–1.164025029	1.451	–0.0014	[237]	$X\ ^1\Sigma_g^+$
(1,–1,1,0)	–1.163485171	2.217	–0.0014	[237]	$X\ ^1\Sigma_g^+$
(0,+1,1,0)	–0.753027184	7.714	1.3813	[238]	$B\ ^1\Sigma_u^+$
(1,–1,0,0)	–0.752850232	1.515	1.4435	[238]	$B\ ^1\Sigma_u^+$

a: L: Spatial angular momentum quantum number; p: parity ($p = (-1)^L$); S_p and S_e: total spin quantum numbers for the protons and the electrons, respectively.
b: Energy obtained from the parameter set used in a recent work [29]. The parameter sets are available in the supporting information of Ref. [217].
c: The virial ratio: $\eta = |1 + \langle\Psi|\hat{V}|\Psi\rangle / (2\langle\Psi|\hat{T}|\Psi\rangle)|$ where $\langle\Psi|\hat{T}|\Psi\rangle$ and $\langle\Psi|\hat{V}|\Psi\rangle$ are the kinetic and the potential energies respectively. In the case of the exact wave function we find $\eta = 0$.
d: Born–Oppenheimer electronic state label. Each energy level given here can be assigned to the lowest-energy vibrational level of the electronic state.

Table 5.2: Electric transition dipole moments obtained in our work for $H_2 = \{p^+, p^+, e^-, e^-\}$. The initial and final states are characterized by their assigned state label and the associated quantum numbers. All results are in atomic units. Only vibrational ground states are considered.

Initial State		Final State			Transition Dipole Moments	
Assignment[a]	(L,p,S_p,S_e)[b]	Assignment[a]	(L,p,S_p,S_e)[b]	E_i-E_f [E_h]	$\lvert\boldsymbol{\mu}_{if}^{(v)}\rvert$	$\lvert\boldsymbol{\mu}_{if}^{(l)}\rvert$
$X\ ^1\Sigma_g^+$	(0,+1,0,0)	$B\ ^1\Sigma_u^+$	(1,–1,0,0)	–0.411174797	0.078415	0.078414
$B\ ^1\Sigma_u^+$	(0,+1,1,0)	$X\ ^1\Sigma_g^+$	(1,–1,1,0)	–0.410457987	0.079802	0.079801

a: Born–Oppenheimer electronic state label. Each energy level given here can be assigned to the lowest-energy vibrational level of the electronic state.
b: L: Spatial angular momentum quantum number; p: parity ($p = (-1)^L$); S_p and S_e: total spin quantum numbers for the protons and the electrons, respectively.

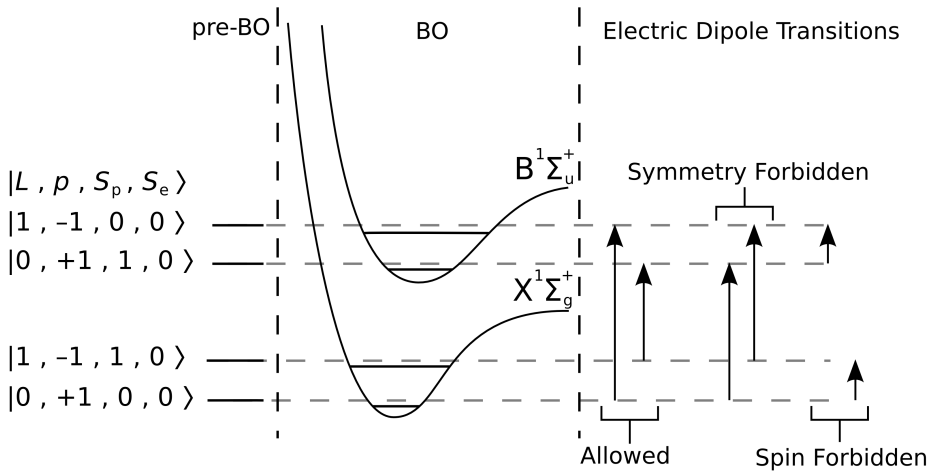

Figure 5.1: Schematic visualization of the spectrum of $H_2 = \{p^+, p^+, e^-, e^-\}$ for the lowest two rovibronic states involved in this chapter. All states are illustrated in terms of the BO and the pre-BO framework. The individual electronic states are designated by their electronic state labels $X\ ^1\Sigma_g^+$ and $B\ ^1\Sigma_u^+$ and the relevant quantum numbers (L: spatial angular momentum state; p: parity $(-1)^L$; S_e: electronic spin state; S_p: proton spin state). Furthermore, all potential transitions are listed and it is designated whether the transitions are allowed or forbidden with respect to electric dipole transitions. Note that the pre-BO energy levels are generally higher than the corresponding BO energy levels.

($B\ ^1\Sigma_u^+$) and therefore suited for the calculation of electric transition dipole moments.

The calculated electric transition dipole moments are listed in table 5.2. The transition dipole moments are presented with a precision that shows the first differing digit. We recognize that the values for $|\boldsymbol{\mu}_{if}^{(l)}|$ and $|\boldsymbol{\mu}_{if}^{(v)}|$ have converged.

5.4 Summary

In this chapter, we presented the expressions for the integrals of the electric transition dipole moments and its squared length. We exploited the simple form of the electric transition dipole operators in LFCC. Integral expressions were presented for the components of the transition dipole in the length and the velocity representation. These two representations are only truly equivalent in the case of the exact wave functions.

We have then calculated the electric transition dipole moments for the H_2 molecule for transitions between the lowest two rovibronic levels of the ortho- and para-H_2. Some sensitivity to the approximation of the wave functions

was observed. Yet this was negligible and we obtained converged results for the electric transition dipole moments.

Furthermore we illustrated the strength of our scheme for the elimination of the translational contribution to any internal molecular property. This scheme was presented in our previous work [81] and allows us to perform pre-BO calculations in LFCC. Previously, a linear combination of the LFCC into a set of TICC, which separates the center-of-mass Cartesian coordinate, had to be performed in order to eliminate any contribution from the overall motion of the system.

6

Ensuring Variational Stability

In this chapter, we focus on the issue of variational stability. Four-component calculations rely on the kinetic-balance condition for variational stability. This condition is well-defined for single fermions [14–22] and has therefore been solely applied to orbital-based methods such as the Dirac–Hartree–Fock approach and relativistic variants of electron-correlation methods [99–101, 106–108, 113, 114, 116, 119, 125–127]. A first solution to the problem of kinetic-balance for explicitly correlated trial wave functions was only recently presented by Pestka and co-workers who have published a series of papers investigating the relativistic helium isotropic series treated as a two-electron system in a central potential [58, 72, 182, 189, 192, 193]. Their solution is an infinite series of transformations of the individual components of the two-electron 16-spinor which is truncated in order to obtain an approximately kinetically balanced trial wave function. Unfortunately, little technical information was provided in Refs. [58, 72, 182, 189, 192, 193] and it remains unclear how such an approximate kinetic-balance condition can be extended to systems containing more than two fermions. Other work focusing on explicitly correlated four-component methods was recently presented by Ten-no and Yamaki who have formulated an explicitly correlated four-component second-order Møller–Plesset perturbation theory using positive-energy-states projection operators in combination with the one-electron kinetic-balance condition [239]. Li and co-workers have studied coalescence conditions for explicitly correlated four-component wave functions [240] but without addressing the issue of kinetic-balance.

As an alternative to the kinetic-balance condition, we investigate a numerical

solution to the problem of variational stability. A modified matrix form of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ was proposed by Schwarz and Mark [129].

6.1 Modified Matrix Form of the Dirac Hamiltonian

Mark and Schwarz have shown that variational collapse is primarily caused by the incompleteness of a finite basis [129]. The problem is related to the identity

$$\langle \boldsymbol{\Psi} | \, \boldsymbol{\sigma} \cdot \boldsymbol{p} \, | \boldsymbol{\Psi} \rangle \, \langle \boldsymbol{\Psi} | \, \boldsymbol{\sigma} \cdot \boldsymbol{p} \, | \boldsymbol{\Psi} \rangle \leq \langle \boldsymbol{\Psi} | \, \boldsymbol{p}^2 \, | \boldsymbol{\Psi} \rangle \quad . \tag{6.1}$$

For an incomplete basis, this relation becomes an inequality and the kinetic energy of the non-relativistic limit is underestimated. Therefore is the total energy underestimated which results in an energy, lower than the ground state, i.e., variational collapse occurs. This problem becomes less severe for a large number of basis functions and the energy obtained for the Dirac hydrogen atom from a calculation using 1000 basis functions is accurate to at least 9 significant digits.

If the basis functions for a one-fermion system have the form

$$\boldsymbol{\phi}_i = \begin{cases} \begin{bmatrix} \boldsymbol{\phi}_i^l \\ \mathbf{0} \end{bmatrix} & \text{if} \quad i < n/2 \\ \\ \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\phi}_i^s \end{bmatrix} & \text{if} \quad i \geq n/2 \end{cases} \tag{6.2}$$

then the matrix form of the $\boldsymbol{\alpha} \cdot \boldsymbol{p}$ operator exhibits a particular super structure:

$$[\boldsymbol{\alpha} \cdot \boldsymbol{p}] = \begin{bmatrix} [0] & [\boldsymbol{\sigma} \cdot \boldsymbol{p}] \\ [\boldsymbol{\sigma} \cdot \boldsymbol{p}] & [0] \end{bmatrix} \quad . \tag{6.3}$$

where the off-diagonal matrices are the matrix representation of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator with the entries

$$[\boldsymbol{\sigma} \cdot \boldsymbol{p}]_{IJ} = \langle \boldsymbol{\phi}_I | \, \boldsymbol{\sigma} \cdot \boldsymbol{p} \, | \boldsymbol{\phi}_J \rangle \quad . \tag{6.4}$$

Schwarz and Mark [129] have proposed a modified matrix form of the $[\boldsymbol{\sigma} \cdot \boldsymbol{p}]$ operator based on the super structure of the matrix form of the $\boldsymbol{\alpha} \cdot \boldsymbol{p}$ operator in

Eq. (6.3). The purpose of this transformation is to correct the non-relativistic limit such that the relation

$$[\boldsymbol{\sigma}\cdot\boldsymbol{p}]^{\mathrm{T}}_{\mathrm{mod}}\,[\boldsymbol{\sigma}\cdot\boldsymbol{p}]_{\mathrm{mod}} = [\boldsymbol{p}^2] \tag{6.5}$$

is fulfilled even in a finite basis. For simplicity, only the one-fermion case is considered here.

Their modified form reads as

$$[\boldsymbol{\sigma}\cdot\boldsymbol{p}]_{\mathrm{mod1}} = [\boldsymbol{\sigma}\cdot\boldsymbol{p}]\left([\boldsymbol{\sigma}\cdot\boldsymbol{p}]^{-1}[\boldsymbol{p}^2][\boldsymbol{\sigma}\cdot\boldsymbol{p}]^{-1}\right)^{1/2} \tag{6.6}$$

where the square braces indicate that all operations are performed with the matrix representations of the operators. Eq. (6.6) minimizes the relative difference between $[\boldsymbol{\sigma}\cdot\boldsymbol{p}]_{\mathrm{mod1}}$ and $[\boldsymbol{\sigma}\cdot\boldsymbol{p}]$. In a footnote, Mark and Schwarz [129] present another transformation which minimizes the absolute difference between $[\boldsymbol{\sigma}\cdot\boldsymbol{p}]_{\mathrm{mod2}}$ and $[\boldsymbol{\sigma}\cdot\boldsymbol{p}]$:

$$[\boldsymbol{\sigma}\cdot\boldsymbol{p}]_{\mathrm{mod2}} = [\boldsymbol{p}^2][\boldsymbol{\sigma}\cdot\boldsymbol{p}]\left([\boldsymbol{\sigma}\cdot\boldsymbol{p}][\boldsymbol{p}^2][\boldsymbol{\sigma}\cdot\boldsymbol{p}]\right)^{-1/2} \tag{6.7}$$

In principle, is is also possible to form the matrix form of $\boldsymbol{\sigma}\cdot\boldsymbol{p}$ as the matrix square-root of the matrix form of $\boldsymbol{p}^2$:

$$[\boldsymbol{\sigma}\cdot\boldsymbol{p}]_{\mathrm{mod3}} = [\boldsymbol{p}^2]^{1/2}\ . \tag{6.8}$$

While this form is calculated most easily, it does not contain information from the unmodified $\boldsymbol{\sigma}\cdot\boldsymbol{p}$ operator.

The modified $[\boldsymbol{\sigma}\cdot\boldsymbol{p}]$ matrix can then be used to form the modified $[\boldsymbol{\alpha}\cdot\boldsymbol{p}]$ matrix as

$$[\boldsymbol{\alpha}\cdot\boldsymbol{p}]_{\mathrm{mod}} = \begin{bmatrix} [0] & [\boldsymbol{\sigma}\cdot\boldsymbol{p}]_{\mathrm{mod}} \\ [\boldsymbol{\sigma}\cdot\boldsymbol{p}]_{\mathrm{mod}} & [0] \end{bmatrix}\ . \tag{6.9}$$

The first two transformations involve the inverse of the $[\boldsymbol{\sigma}\cdot\boldsymbol{p}]$ matrix. This step is prone to severe numerical instabilities and requires great care when generating a parameter set. This problem can partially be solved using an increased numerical precision, e.g., using the GMP library [241].

The square root of a matrix $\boldsymbol{X}$ can be calculated from the diagonal eigenvalue matrix $\boldsymbol{\Omega}$ and ortho-normal eigenvectors $\boldsymbol{Q}$ as

$$\boldsymbol{X}^{1/2} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{\Omega}^{1/2}\boldsymbol{Q} \tag{6.10}$$

where $\mathbf{\Omega}^{1/2}$ is a diagonal matrix containing the square-root of the eigenvalues as elements. However, the evaluation of $\boldsymbol{Q}$ is another source of numerical problems. This issue is not so easily solved. Extremely high numerical precision is required to ensure numerical stability. This is not a problem of the implementation or the algorithm. We have tested the MPACK library [242] (an implementation of LAPACK and BLAS based on GMP) and the eigen package [241] using MPFR [243] for increased numerical accuracy. They rely on the same eigensolver (QR algorithm [244–246]) and yield identical results.

Furthermore, we implemented a custom eigensolver based on the divide and conquer algorithm according to Gu and Eisenstat [247] using the MPFR library and the eigen package. It yields identical results as the QR algorithm. In addition to the increased numerical precision, a simple test

$$\max |[\boldsymbol{X}] - [\boldsymbol{X}]^{1/2}[\boldsymbol{X}]^{1/2}| \leq \epsilon \ , \tag{6.11}$$

where ϵ is some threshold to be specified by the user, is performed and parameter sets where Eq. (6.11) is not fulfilled are discarded in the process of stochastic sampling.

Calculations using the numerically stabilized framework still exhibit prolapse. This is independent from the modified form of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator. All of them yield results for the Dirac hydrogen atom with relativistic effects about ten times as high as compared to the analytical solution.

6.2 Derivation of the One-Electron Kinetic-Balance Condition

An effective means of dealing with the problem of variational collapse is the kinetic-balance condition [14–22] which relates the large and the small component of the one-fermion eigenfunction:

$$\boldsymbol{\psi}^s(\boldsymbol{r}) \approx \frac{\boldsymbol{\sigma} \cdot \boldsymbol{p}}{2mc} \boldsymbol{\psi}^l(\boldsymbol{r}) \ . \tag{6.12}$$

The derivation of this relation is straightforward. The Dirac eigenvalue problem

$$(\boldsymbol{h}_D - E)\boldsymbol{\psi}(\boldsymbol{r}) = 0 \tag{6.13}$$

leads to a set of two linear equations for the two-component parts of the 4-spinor of Eq. (3.4). After the energy spectrum has been shifted by $-mc^2$, this system of equations reads

$$(V - E)\boldsymbol{\psi}^l(\boldsymbol{r}) + c\boldsymbol{\sigma} \cdot \boldsymbol{p}\ \boldsymbol{\psi}^s(\boldsymbol{r}) = 0\ , \tag{6.14}$$

$$(V - E - 2mc^2)\boldsymbol{\psi}^s(\boldsymbol{r}) + c\boldsymbol{\sigma} \cdot \boldsymbol{p}\ \boldsymbol{\psi}^l(\boldsymbol{r}) = 0\ . \tag{6.15}$$

We only need one of the two equations to relate the small component to the large one. Since the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ has no multiplicative inverse, it is more convenient to choose the second equation in order to obtain an expression for $\boldsymbol{\psi}^s(\boldsymbol{r})$. After rearranging the terms, we obtain the exact relation for Eq. (6.15)

$$\boldsymbol{\psi}^s(\boldsymbol{r}) = \frac{c\boldsymbol{\sigma} \cdot \boldsymbol{p}}{(E - V + 2mc^2)}\boldsymbol{\psi}^l(\boldsymbol{r})\ . \tag{6.16}$$

This relation depends on the energy of the system which is not known a priori but is one of the desired results of the problem. Eq. (6.16) can therefore not be applied to our problem. However, due to the large value of mc^2 and the fact that we are interested in systems for which E and V are comparatively small, we can introduce the approximation

$$E - V + 2mc^2 \approx 2mc^2\ . \tag{6.17}$$

Inserting this relation into Eq. (6.16) yields the kinetic-balance condition in Eq. (6.12).

Obeying Eq. (6.12) for basis-set expansions yields a variational stable parametrization of the trial wave function. For convenience, we denote the kinetically balanced components as

$$\boldsymbol{\psi}(\boldsymbol{r}) \approx \begin{bmatrix} |l\rangle \\ |s\rangle \end{bmatrix} = \begin{bmatrix} 1 \\ \dfrac{\boldsymbol{\sigma} \cdot \boldsymbol{p}}{2mc} \end{bmatrix} \boldsymbol{\psi}^l(\boldsymbol{r}) \tag{6.18}$$

Eq. (6.12) may also be formulated in terms of the one-fermion model spaces [18, 71, 72]

$$|l\rangle \in \mathcal{H}^l \quad \text{and} \quad |s\rangle \in (\boldsymbol{\sigma} \cdot \boldsymbol{p})\mathcal{H}^l \subset \mathcal{H}^s\ . \tag{6.19}$$

The kinetic-balance condition in Eq. (6.12) also ensures the correct non-relativistic (NR) limit for $c \to \infty$. Forming the Rayleigh coefficient according to Eq. (3.29) and taking the limit $c \to \infty$ leads to the non-relativistic Schrödinger energy:

$$E_{\mathrm{NR}} = \lim_{c\to\infty} \frac{\langle \boldsymbol{\Psi} | \boldsymbol{h}_D | \boldsymbol{\Psi} \rangle}{\langle \boldsymbol{\Psi} | \boldsymbol{\Psi} \rangle} = \frac{\left\langle l \left| \dfrac{p^2}{2m} + V \right| l \right\rangle}{\langle l | l \rangle}\ . \tag{6.20}$$

For our purposes here, it is important to realize that the small component is imaginary

$$|s\rangle = -\langle s| \tag{6.21}$$

if $|l\rangle$ is chosen to be real, since the momentum operator in the kinetic-balance condition is imaginary.

6.3 Partitioning of the Wave Function

The wave function for N *non*-interacting fermions can be constructed as the direct product of one-fermion 4-spinors $\boldsymbol{\psi}_i(\boldsymbol{r}_i)$,

$$\boldsymbol{\Psi}(\boldsymbol{r}) = \boldsymbol{\psi}_1(\boldsymbol{r}_1) \otimes \ldots \otimes \boldsymbol{\psi}_i(\boldsymbol{r}_i) \otimes \ldots \otimes \boldsymbol{\psi}_N(\boldsymbol{r}_N) \ , \tag{6.22}$$

where we leave aside the antisymmetrization for the sake of brevity. Now, $\boldsymbol{r} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)^{\mathrm{T}}$ collects all N one-fermion coordinates. In the case of two fermions, we have the direct product of two basis states

$$\boldsymbol{\psi}_1(\boldsymbol{r}_1) \otimes \boldsymbol{\psi}_2(\boldsymbol{r}_2) = \begin{bmatrix} \psi_1^{l1}(\boldsymbol{r}_1) \\ \psi_1^{l2}(\boldsymbol{r}_1) \\ \psi_1^{s1}(\boldsymbol{r}_1) \\ \psi_1^{s2}(\boldsymbol{r}_1) \end{bmatrix} \otimes \begin{bmatrix} \psi_2^{l1}(\boldsymbol{r}_2) \\ \psi_2^{l2}(\boldsymbol{r}_2) \\ \psi_2^{s1}(\boldsymbol{r}_2) \\ \psi_2^{s2}(\boldsymbol{r}_2) \end{bmatrix} = \begin{bmatrix} \psi_1^{l1}(\boldsymbol{r}_1)\psi_2^{l1}(\boldsymbol{r}_2) \\ \psi_1^{l1}(\boldsymbol{r}_1)\psi_2^{l2}(\boldsymbol{r}_2) \\ \psi_1^{l1}(\boldsymbol{r}_1)\psi_2^{s1}(\boldsymbol{r}_2) \\ \psi_1^{l1}(\boldsymbol{r}_1)\psi_2^{s2}(\boldsymbol{r}_2) \\ \psi_1^{l2}(\boldsymbol{r}_1)\psi_2^{l1}(\boldsymbol{r}_2) \\ \psi_1^{l2}(\boldsymbol{r}_1)\psi_2^{l2}(\boldsymbol{r}_2) \\ \vdots \\ \psi_1^{s2}(\boldsymbol{r}_1)\psi_2^{s2}(\boldsymbol{r}_2) \end{bmatrix} \ . \tag{6.23}$$

The superscript indicates the element of the one-fermion four-spinor: The letters l and s indicate the large or the small 2-spinor as before. The number attached to these letters indicates the element of a 2-spinor. For instance, the elements of the large-component 2-spinor are then

$$\psi_1^l(\boldsymbol{r}_1) = \begin{bmatrix} \psi_1^{l1}(\boldsymbol{r}_1) \\ \psi_1^{l2}(\boldsymbol{r}_1) \end{bmatrix} \ . \tag{6.24}$$

An N-fermion wave function for arbitrary fermions may be constructed to be consistent with the model space

$$\mathcal{H}^{(N)} = \mathcal{H}^{l\ldots l} \oplus \ldots \oplus \mathcal{H}^{\lambda_1 \ldots \lambda_N} \oplus \ldots \oplus \mathcal{H}^{s\ldots s}. \tag{6.25}$$

where each $\mathcal{H}^{\lambda_1 \dots \lambda_N}$ is constructed from the one fermion model spaces

$$\mathcal{H}^{\lambda_1 \dots \lambda_N} = \mathcal{H}^{\lambda_1} \otimes \dots \otimes \mathcal{H}^{\lambda_N} \tag{6.26}$$

with $\lambda_1, \dots, \lambda_N \in \{l, s\}$. The highlighted vector components in Eq. (6.23) are those contained within the model space $\mathcal{H}^{ll}$. We recognize that the wave function in Eq. (6.22) and the model space in Eq. (6.25) are not compatible since it is not possible to partition Eq. (6.22) in terms of the one-fermion model spaces. However, we can reorder the spinor elements of the wave function as

$$\boldsymbol{P}^{\mathrm{T}} \boldsymbol{\Psi}(\boldsymbol{r}) = \boldsymbol{\psi}_1(\boldsymbol{r}_1) \boxtimes \dots \boxtimes \boldsymbol{\psi}_i(\boldsymbol{r}_i) \boxtimes \dots \boxtimes \boldsymbol{\psi}_N(\boldsymbol{r}_N) \tag{6.27}$$

where $\boxtimes$ is the Tracy-Singh product and $\boldsymbol{P}$ is a permutation matrix (see Eq. (B.4) and further detaills in the appendix B.1). Then, our two-spinor example reads

$$\boldsymbol{\psi}_1(\boldsymbol{r}_1) \boxtimes \boldsymbol{\psi}_2(\boldsymbol{r}_2) = \begin{bmatrix} \boldsymbol{\psi}_1^l(\boldsymbol{r}_1) \otimes \boldsymbol{\psi}_2^l(\boldsymbol{r}_2) \\ \boldsymbol{\psi}_1^l(\boldsymbol{r}_1) \otimes \boldsymbol{\psi}_2^s(\boldsymbol{r}_2) \\ \boldsymbol{\psi}_1^s(\boldsymbol{r}_1) \otimes \boldsymbol{\psi}_2^l(\boldsymbol{r}_2) \\ \boldsymbol{\psi}_1^s(\boldsymbol{r}_1) \otimes \boldsymbol{\psi}_2^s(\boldsymbol{r}_2) \end{bmatrix} = \begin{bmatrix} \psi_1^{l1}(\boldsymbol{r}_1)\psi_2^{l1}(\boldsymbol{r}_2) \\ \psi_1^{l1}(\boldsymbol{r}_1)\psi_2^{l2}(\boldsymbol{r}_2) \\ \psi_1^{l2}(\boldsymbol{r}_1)\psi_2^{l1}(\boldsymbol{r}_2) \\ \psi_1^{l2}(\boldsymbol{r}_1)\psi_2^{l2}(\boldsymbol{r}_2) \\ \psi_1^{l1}(\boldsymbol{r}_1)\psi_2^{s1}(\boldsymbol{r}_2) \\ \psi_1^{l1}(\boldsymbol{r}_1)\psi_2^{s2}(\boldsymbol{r}_2) \\ \vdots \\ \psi_1^{s2}(\boldsymbol{r}_1)\psi_2^{s2}(\boldsymbol{r}_2) \end{bmatrix} . \tag{6.28}$$

The vector components highlighted in Eq. (6.28) are those contained within the $\mathcal{H}^{ll}$ model space as in Eq. (6.23). We see that the wave function in Eq. (6.28) can be partitioned such that the individual components are part of the different model spaces in Eq. (6.25),

$$\left[\boldsymbol{P}^{\mathrm{T}} \boldsymbol{\Psi}(\boldsymbol{r})\right]^{\lambda_1 \dots \lambda_N} = \boldsymbol{\psi}_1^{\lambda_1}(\boldsymbol{r}_1) \otimes \dots \otimes \boldsymbol{\psi}_i^{\lambda_i}(\boldsymbol{r}_i) \otimes \dots \otimes \boldsymbol{\psi}_N^{\lambda_N}(\boldsymbol{r}_N) \ , \tag{6.29}$$

where $\lambda_1, \dots, \lambda_N \in \{l, s\}$ as in Eq. (6.26).

The Hamiltonian is transformed accordingly (cf. Eq. (B.3) in the appendix)

$$\begin{aligned} \boldsymbol{H}_{DTS}^{(N)} &= \boldsymbol{P}^{\mathrm{T}} \boldsymbol{H}_{D}^{(N)} \boldsymbol{P} \\ &= \sum_{i=1}^{N} \boldsymbol{P}^{\mathrm{T}} \left(\boldsymbol{h}_D(i) + \boldsymbol{W}\right) \boldsymbol{P} = \sum_{i=1}^{N} \boldsymbol{h}_{DTS}(i) + \boldsymbol{P}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{P} \end{aligned} \tag{6.30}$$

with

$$\begin{aligned}\boldsymbol{h}_{DTS}(i) &= \boldsymbol{P}^{\mathrm{T}}\boldsymbol{h}_D(i)\boldsymbol{P} \\ &= \mathbf{1}_4(1) \boxtimes \cdots \boxtimes \mathbf{1}_4(i-1) \boxtimes \boldsymbol{h}_D(\boldsymbol{r}_i) \boxtimes \mathbf{1}_4(i+1) \boxtimes \cdots \boxtimes \mathbf{1}_4(N) \end{aligned} \quad (6.31)$$

where all matrices are partitioned into 2×2 blocks. The potential-energy operator

$$\boldsymbol{W} = \sum_{i<j}^{N} \boldsymbol{g}(i,j) \ , \quad (6.32)$$

which describes the pairwise interaction among the fermions, will be invariant under this transformation if the instantaneous Coulomb interaction only is considered as it is a diagonal matrix with identical entries. The situation is more complicated when magnetic interactions are taken into account. An N-fermion wave function for 1/2-fermions can then be partitioned in terms of the model space into 2^N components each of dimension 2^N,

$$\boldsymbol{\Psi}(\boldsymbol{r}) = \begin{bmatrix} \boldsymbol{\Psi}^{l\ldots l}(\boldsymbol{r}) \\ \cdots \\ \boldsymbol{\Psi}^{\lambda_1\ldots\lambda_N}(\boldsymbol{r}) \\ \cdots \\ \boldsymbol{\Psi}^{s\ldots s}(\boldsymbol{r}) \end{bmatrix} \ . \quad (6.33)$$

Note that a related reordering of the Hamiltonian similar to Eq. (6.30) is key for the quaternion formulation of four-component self-consistent filed algorithms [248].

6.4 Exact Two-Particle Kinetic-Balance Condition

In this section, we derive the kinetic-balance condition for explicitly correlated basis functions for a system of two fermions. According to Eq. (6.25) the model space takes the form

$$\mathcal{H}^{(2)} = \mathcal{H}^{ll} \oplus \mathcal{H}^{ls} \oplus \mathcal{H}^{sl} \oplus \mathcal{H}^{ss} \quad (6.34)$$

where the four model spaces

$$\mathcal{H}^{ll} = \mathcal{H}^{l} \otimes \mathcal{H}^{l} \quad (6.35)$$
$$\mathcal{H}^{ls} = \mathcal{H}^{l} \otimes \mathcal{H}^{s} \quad (6.36)$$
$$\mathcal{H}^{sl} = \mathcal{H}^{s} \otimes \mathcal{H}^{l} \quad (6.37)$$
$$\mathcal{H}^{ss} = \mathcal{H}^{s} \otimes \mathcal{H}^{s} \quad (6.38)$$

are formed from the single-fermion model spaces $\mathcal{H}^l$ and $\mathcal{H}^s$. Each model space in Eqs. (6.35) – (6.38) is assigned to one of the four components in the wave function. The Dirac Hamiltonian has to respect the Tracy-Singh product (see Eq. (B.1) in the appendix) to match the partition of the wave function according to Eq. (6.33). We then obtain the following block structure for the two-fermion Hamiltonian defined in Eq. (6.30):

$$\boldsymbol{H}^{(2)}_{DTS}(\boldsymbol{r}_1,\boldsymbol{r}_2) = \begin{bmatrix} \boldsymbol{V} & c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) & 0 \\ c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & \boldsymbol{V}-2m_2c^2 & 0 & c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) \\ c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) & 0 & \boldsymbol{V}-2m_1c^2 & c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) \\ 0 & c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) & c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & \boldsymbol{V}-2m_{12}c^2 \end{bmatrix} \tag{6.39}$$

where we have introduced an energy shift of the whole spectrum by a factor of $-m_{12}c^2$ with $m_{12} = m_1 + m_2$ and introduced $\boldsymbol{V} = \boldsymbol{V}(\boldsymbol{r}_1) + \boldsymbol{V}(\boldsymbol{r}_2) + \boldsymbol{W}$ as the total potential-energy operator. Also, we absorbed the direct products into $\boldsymbol{\sigma}_i$ as

$$\boldsymbol{\sigma}_1 = (\boldsymbol{\sigma}_x\otimes\mathbf{1}_2, \boldsymbol{\sigma}_y\otimes\mathbf{1}_2, \boldsymbol{\sigma}_z\otimes\mathbf{1}_2)^{\mathrm{T}} \ , \tag{6.40}$$

and

$$\boldsymbol{\sigma}_2 = (\mathbf{1}_2\otimes\boldsymbol{\sigma}_x, \mathbf{1}_2\otimes\boldsymbol{\sigma}_y, \mathbf{1}_2\otimes\boldsymbol{\sigma}_z)^{\mathrm{T}} \ . \tag{6.41}$$

The idea of kinetic-balance is to relate the small-component one-fermion model spaces to their large-component one-fermion model spaces in the eigenvalue problem

$$\left(\boldsymbol{H}^{(2)}_{DTS} - E\right)\boldsymbol{\Psi}(\boldsymbol{r}_1,\boldsymbol{r}_2) = 0 \ . \tag{6.42}$$

This leads to a system of four equations, according to the Eqs. (6.14) and (6.15):

$$0 = (\boldsymbol{V}-E)\boldsymbol{\Psi}^{ll} + c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\boldsymbol{\Psi}^{ls} + c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)\boldsymbol{\Psi}^{sl} \tag{6.43}$$

$$0 = c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\boldsymbol{\Psi}^{ll} + (\boldsymbol{V}-E-2m_2c^2)\boldsymbol{\Psi}^{ls} + c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)\boldsymbol{\Psi}^{ss} \tag{6.44}$$

$$0 = c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)\boldsymbol{\Psi}^{ll} + (\boldsymbol{V}-E-2m_1c^2)\boldsymbol{\Psi}^{sl} + c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\boldsymbol{\Psi}^{ss} \tag{6.45}$$

$$0 = c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)\boldsymbol{\Psi}^{ls} + c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\boldsymbol{\Psi}^{sl} + (\boldsymbol{V}-E-2m_{12}c^2)\boldsymbol{\Psi}^{ss} \tag{6.46}$$

where we have suppressed the coordinate dependence of the 4-spinors and will continue to do so where appropriate for the sake of brevity. We have to eliminate one of these four equations because we search for relation between the four components of the wave function which we can then apply as a constraint for choosing explicitly correlated basis functions. As in the case of

a single fermion, we eliminate the energy E from the equations using the approximate relation

$$2m_ic^2 + E - V \approx 2m_ic^2 \tag{6.47}$$

where $m_i \in \{m_1, m_2, m_1 + m_2\}$. We eliminate the first equation, Eq. (6.43), from the system of equations since it is the only equation where $2m_ic^2$ does not occur so that Eq. (6.47) cannot be applied. After applying Eq. (6.47) to Eqs. (6.44) – (6.46), we find the following relations among the four components of the two-fermion wave function:

$$0 \approx c(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)\boldsymbol{\Psi}^{ll} - 2m_2c^2\boldsymbol{\Psi}^{ls} + c(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)\boldsymbol{\Psi}^{ss} \tag{6.48}$$

$$0 \approx c(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)\boldsymbol{\Psi}^{ll} - 2m_1c^2\boldsymbol{\Psi}^{sl} + c(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)\boldsymbol{\Psi}^{ss} \tag{6.49}$$

$$0 \approx c(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)\boldsymbol{\Psi}^{ls} + c(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)\boldsymbol{\Psi}^{sl} - 2m_{12}c^2\boldsymbol{\Psi}^{ss} \ . \tag{6.50}$$

The matrix form of this under-determined system of linear equations can be interpreted as the augmented form of a linear system with a unique solution:

$$\boldsymbol{A} = \left[\begin{array}{ccc|c} (\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2) & -2m_2c & 0 & -(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1) \\ (\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1) & 0 & -2m_1c & -(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2) \\ 0 & (\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1) & (\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2) & 2m_{12}c \end{array}\right] \begin{array}{l}\{1\}\\ \{2\}\\ \{3\}\end{array} \tag{6.51}$$

The augmented form of linear systems and row reduction are explained in more detail in section B.2 in the appendix. The number in curly brackets on the right-hand side of every row indicates the line number. It will be used to express the manipulations in the row reduction below.

There is no row-reduced echelon form for the augmented form in Eq. (6.51). The lack of a multiplicative inverse of the differential operator prohibits us from setting the leading element of each row of the row-reduced echelon form to 1 (see Eq. (B.8) in the appendix) and therefore to relate $\Psi^{ll}(\boldsymbol{r}_1, \boldsymbol{r}_2)$, $\Psi^{ls}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ and $\Psi^{sl}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ to $\Psi^{ss}(\boldsymbol{r}_1, \boldsymbol{r}_2)$. However, we are able to find a similar form with pairwise relations between $\Psi^{ss}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ and the other three components. We now present the individual steps necessary to obtain this modified row-reduced echelon form:

1. Insert $(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)\{1\} - (\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)\{2\}$ into $\{2\}$:

$$\left[\begin{array}{ccc|c} (\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2) & -2m_2c & 0 & -(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1) \\ 0 & -2m_2c(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1) & 2m_1c(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2) & \boldsymbol{p}_2^2 - \boldsymbol{p}_1^2 \\ 0 & (\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1) & (\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2) & 2m_{12}c \end{array}\right] \begin{array}{l}\{1\}\\ \{2\}\\ \{3\}\end{array}$$

2. Insert $\{2\} + 2m_2c\{3\}$ into $\{3\}$:

$$\left[\begin{array}{ccc|c} (\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & -2m_2c & 0 & -(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) \\ 0 & -2m_2c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) & 2m_1c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & \boldsymbol{p}_2^2-\boldsymbol{p}_1^2 \\ 0 & 0 & 2m_{12}c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & \boldsymbol{p}_2^2-\boldsymbol{p}_1^2+4m_2m_{12}c^2 \end{array}\right] \begin{array}{c}\{1\}\\ \{2\}\\ \{3\}\end{array}$$

3. Insert $-\dfrac{m_{12}}{m_2}\{2\} + \dfrac{m_1}{m_2}\{3\}$ into $\{2\}$:

$$\left[\begin{array}{ccc|c} (\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & -2m_2c & 0 & -(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) \\ 0 & 2m_{12}c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) & 0 & \boldsymbol{p}_1^2-\boldsymbol{p}_2^2+4m_1m_{12}c^2 \\ 0 & 0 & 2m_{12}c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & \boldsymbol{p}_2^2-\boldsymbol{p}_1^2+4m_2m_{12}c^2 \end{array}\right] \begin{array}{c}\{1\}\\ \{2\}\\ \{3\}\end{array}$$

4. Insert $(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)m_{12}\{1\} + m_2\{2\}$ into $\{1\}$:

$$\left[\begin{array}{ccc|c} m_{12}(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) & 0 & 0 & -m_1\boldsymbol{p}_1^2-m_2\boldsymbol{p}_2^2+4m_1m_{12}c^2 \\ 0 & 2m_{12}c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1) & 0 & \boldsymbol{p}_1^2-\boldsymbol{p}_2^2+4m_1m_{12}c^2 \\ 0 & 0 & 2m_{12}c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2) & \boldsymbol{p}_2^2-\boldsymbol{p}_1^2+4m_2m_{12}c^2 \end{array}\right] \begin{array}{c}\{1\}\\ \{2\}\\ \{3\}\end{array}$$

We instantly recognize that we have now a set of simple pairwise relations between $\boldsymbol{\Psi}^{ss}(\boldsymbol{r}_1,\boldsymbol{r}_2)$ and the other three components

$$-(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)m_{12}\boldsymbol{\Psi}^{ll} = \left(m_1\boldsymbol{p}_1^2+m_2\boldsymbol{p}_2^2-4m_1m_2m_{12}c^2\right)\boldsymbol{\Psi}^{ss} \tag{6.52}$$

$$-2c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)m_{12}\boldsymbol{\Psi}^{ls} = \left(\boldsymbol{p}_2^2-\boldsymbol{p}_1^2-4m_1m_{12}c^2\right)\boldsymbol{\Psi}^{ss} \tag{6.53}$$

$$-2c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)m_{12}\boldsymbol{\Psi}^{sl} = \left(\boldsymbol{p}_1^2-\boldsymbol{p}_2^2-4m_2m_{12}c^2\right)\boldsymbol{\Psi}^{ss} \tag{6.54}$$

and note the similarity of the operators acting on the components on the left hand sides of the equations. Forming the least common multiple from these operators, we can introduce a 4-spinor $\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2)$ related to the $\boldsymbol{\Psi}^{ss}(\boldsymbol{r}_1,\boldsymbol{r}_2)$ component

$$\boldsymbol{\Psi}^{ss}(\boldsymbol{r}_1,\boldsymbol{r}_2) = -2cm_{12}(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2)\ , \tag{6.55}$$

insert it into Eqs. (6.52) – (6.54) and eliminate identical terms on both sides. Instead of relating the upper component to the lower component, we relate all four component to some spinor $\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2)$ having the same dimension as a single component:

$$|ll\rangle = U_{ll}\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2) = \left(m_1\boldsymbol{p}_1^2+m_2\boldsymbol{p}_2^2-4m_1m_2m_{12}c^2\right)\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2) \tag{6.56}$$

$$|ls\rangle = U_{ls}\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2) = \frac{(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)}{2c}\left(\boldsymbol{p}_2^2-\boldsymbol{p}_1^2-4m_1m_{12}c^2\right)\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2) \tag{6.57}$$

$$|sl\rangle = U_{sl}\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2) = \frac{(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)}{2c}\left(\boldsymbol{p}_1^2-\boldsymbol{p}_2^2-4m_2m_{12}c^2\right)\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2) \tag{6.58}$$

$$|ss\rangle = U_{ss}\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2) = -m_{12}(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2)\ . \tag{6.59}$$

Here, we have introduced the short-hand notation for balanced components equivalent to the one-fermion case. The physical role of $\boldsymbol{\Theta}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ will become clear when we study the non-relativistic limit (see below).

$\boldsymbol{\Psi}^{ss}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ is for any type of function uniquely defined by $\boldsymbol{\Theta}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ up to a constant factor, i.e., the constant of integration, due to the differential operators involved. For square-integrable functions, this constant factor is zero. Hence, cancellation of differential operators is not a problem and all components are uniquely determined by $\boldsymbol{\Theta}(\boldsymbol{r}_1, \boldsymbol{r}_2)$.

Eqs. (6.56) – (6.59) show that there is some connection to the one-fermion kinetic-balance condition. The highest-order terms in c in each of the four relations are those one would obtain from the one-fermion kinetic-balance condition applied for both fermions subsequently.

We can now analyze the real and imaginary parts of the four components. Because of the odd number of $(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)$ and $(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)$ operators acting on $|sl\rangle$ and $|ls\rangle$, respectively, we find that these two components are imaginary and because of the even number of $(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)$ and $(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)$ operators acting on $|ll\rangle$ and $|ss\rangle$, respectively, we find that these two components are real.

Last, we consider fermion exchange symmetry for the two identical fermions leading to the relations [58]

$$\boldsymbol{\Psi}^{ll}(\boldsymbol{r}_1, \boldsymbol{r}_2) = -\boldsymbol{\Psi}^{ll}(\boldsymbol{r}_2, \boldsymbol{r}_1) \tag{6.60}$$

$$\boldsymbol{\Psi}^{ls}(\boldsymbol{r}_1, \boldsymbol{r}_2) = -\boldsymbol{\Psi}^{sl}(\boldsymbol{r}_2, \boldsymbol{r}_1) \tag{6.61}$$

$$\boldsymbol{\Psi}^{ss}(\boldsymbol{r}_1, \boldsymbol{r}_2) = -\boldsymbol{\Psi}^{ss}(\boldsymbol{r}_2, \boldsymbol{r}_1) \tag{6.62}$$

which have to be fulfilled in addition to the relations in Eqs. (6.56) – (6.59). $\boldsymbol{\Theta}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ is antisymmetrized before the components are constructed according to Eqs. (6.56) – (6.59) because the operators $(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)$ and $(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)$ do not commute with the permutation operator which exchanges fermions 1 and 2.

6.5 The Non-Relativistic Limit

The one-fermion kinetic-balance condition yields the correct non-relativistic limit for $c \to \infty$. This is a key requirement ensuring variational stability. We therefore require any kinetic-balance condition for more than one fermion to yield the correct non-relativistic limit.

Finding the non-relativistic limit for the one-fermion case is fairly trivial. For the two-fermion kinetic-balance condition, this is somewhat more involved.

In order to find the correct limit, we rely on l'Hôpital's rule for limits,

$$\lim_{x\to y}\frac{f(x)}{g(x)}=\lim_{x\to y}\frac{f'(x)}{g'(x)}. \tag{6.63}$$

where $f'(x)$ and $g'(x)$ are the derivatives of $f(x)$ and $g(x)$ with respect to x and y is the limiting value of x.

The non-relativistic limit of the two-fermion total energy for an kinetically balanced wave function according to Eqs. (6.56) – (6.59), can be taken as a limiting case of the Rayleigh coefficient

$$E_{\mathrm{NR}}[\boldsymbol{H}^{(2)}_{DTS},\boldsymbol{\Theta}]=\lim_{c\to\infty}\frac{\langle\boldsymbol{\Psi}|\,\boldsymbol{H}^{(2)}_{DTS}\,|\boldsymbol{\Psi}\rangle}{\langle\boldsymbol{\Psi}\,|\boldsymbol{\Psi}\rangle}$$

$$=\lim_{c\to\infty}\frac{\langle ll|\,c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)\,|ls\rangle+\langle ll|\,c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\,|sl\rangle+\langle\boldsymbol{\Psi}|\,\boldsymbol{V}\,|\boldsymbol{\Psi}\rangle}{\langle\boldsymbol{\Psi}\,|\boldsymbol{\Psi}\rangle}\ . \tag{6.64}$$

We now apply l'Hôpital's rule to Eq. (6.64) by taking the fourth-order derivative with respect to c of both the nominator and denominator:

$$E_{NR}[\boldsymbol{H}^{(2)}_{DTS},\boldsymbol{\Theta}]=\lim_{c\to\infty}\frac{\dfrac{\partial^4}{\partial c^4}\langle ll|\,c(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)\,|ls\rangle+\langle ll|\,c(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\,|sl\rangle+\langle\boldsymbol{\Psi}|\,\boldsymbol{V}\,|\boldsymbol{\Psi}\rangle}{\dfrac{\partial^4}{\partial c^4}\langle\boldsymbol{\Psi}\,|\boldsymbol{\Psi}\rangle}$$

$$=\lim_{c\to\infty}\frac{\langle\boldsymbol{\Theta}|\,192m_{12}^2m_1m_2^2\boldsymbol{p}_1^2+192m_{12}^2m_1^2m_2\boldsymbol{p}_2^2\,|\boldsymbol{\Theta}\rangle}{384m_{12}^2m_1^2m_2^2\,\langle\boldsymbol{\Theta}\,|\boldsymbol{\Theta}\rangle}$$

$$+\frac{\langle\boldsymbol{\Theta}|\,384m_{12}^2m_1^2m_2^2\boldsymbol{V}+\mathcal{O}(c^{-2})\,|\boldsymbol{\Theta}\rangle}{384m_{12}^2m_1^2m_2^2\,\langle\boldsymbol{\Theta}\,|\boldsymbol{\Theta}\rangle}\ . \tag{6.65}$$

Only the potential-energy term still contains contributions depending on c. These contributions are of order c^{-2} and higher. When taking the limit, they are all zero and we find the limit to be a simpler Rayleigh factor depending on $\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2)$

$$E_{\mathrm{NR}}=\frac{\left\langle\boldsymbol{\Theta}\left|\dfrac{\boldsymbol{p}_1^2}{2m_1}+\dfrac{\boldsymbol{p}_2^2}{2m_2}+\boldsymbol{V}\right|\boldsymbol{\Theta}\right\rangle}{\langle\boldsymbol{\Theta}\,|\boldsymbol{\Theta}\rangle}\ , \tag{6.66}$$

which is clearly the Schrödinger energy and therefore the correct non-relativistic limit is obtained. The limit also identifies $\boldsymbol{\Theta}(\boldsymbol{r}_1,\boldsymbol{r}_2)$ as the non-relativistic two-fermion Schrödinger wave function.

From Eq. (6.65) we see that different components define the non-relativistic limit for the kinetic-energy part and the potential-energy part. For the kinetic energy, the non-relativistic limit is defined by the components $|ll\rangle$, $|ls\rangle$, and $|sl\rangle$. The non-relativistic limit of the potential-energy is only defined by the $|ll\rangle$ component. The $|ss\rangle$ component is not directly involved in defining the non-relativistic limit.

Only the leading terms in c of the three components define the non-relativistic limit since we have used l'Hôpital's rule. The leading terms are

$$|ll\rangle\,(c^2) = -4m_1 m_2 m_{12} c^2 \ , \tag{6.67}$$

$$|ls\rangle\,(c) = -2m_1 m_{12} c(\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2) \tag{6.68}$$

and

$$|sl\rangle\,(c) = -2m_2 m_{12} c(\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1) \ . \tag{6.69}$$

We note the similarity to Eq. (6.12) and realize that the non-relativistic limit is determined by the one-fermion kinetic-balance condition. The one-fermion kinetic-balance condition may not be sufficient to ensure variational stability [18, 71, 72]. Then, the non-relativistic limit will not be a sufficient, albeit necessary condition for variational stability.

6.6 Kinetic-Balance Condition for more than Two Fermions

The derivation presented in section 6.4 can also be applied to systems with more than two fermions. How this can be achieved for the result obtained by Pestka and co-workers [18, 71, 72] is not obvious and was not discussed in their papers. In our ansatz, we obtain rather lengthy expressions for three fermions, which we refrain from presenting explicitly for the sake of brevity. The resulting expressions can, however, be expanded into a polynomial with respect to c. The individual terms $d_i^{(3)}(c)$ feature an important property

$$d_i^{(3)}(c) = k_i(m_1, m_2, m_3) \times c^{(6-u-v-w)} (\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)^u (\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)^v (\boldsymbol{\sigma}_3 \cdot \boldsymbol{p}_3)^w \tag{6.70}$$

where the positive semi-definite exponents u, v and w obey the constraints $0 \leq (u + v + w) \leq 7$ and we have

$$\boldsymbol{\sigma}_1 = (\boldsymbol{\sigma}_x \otimes \mathbf{1}_4, \boldsymbol{\sigma}_y \otimes \mathbf{1}_4, \boldsymbol{\sigma}_z \otimes \mathbf{1}_4)^{\mathrm{T}} \ , \tag{6.71}$$

$$\boldsymbol{\sigma}_2 = (\mathbf{1}_2 \otimes \boldsymbol{\sigma}_x \otimes \mathbf{1}_2, \mathbf{1}_2 \otimes \boldsymbol{\sigma}_y \otimes \mathbf{1}_2, \mathbf{1}_2 \otimes \boldsymbol{\sigma}_z \otimes \mathbf{1}_2)^{\mathrm{T}} \ , \tag{6.72}$$

$$\boldsymbol{\sigma}_3 = (\mathbf{1}_4 \otimes \boldsymbol{\sigma}_x, \mathbf{1}_4 \otimes \boldsymbol{\sigma}_y, \mathbf{1}_4 \otimes \boldsymbol{\sigma}_z)^{\mathrm{T}} \ . \tag{6.73}$$

The multiplicative prefactors $k_i(m_1, m_2, m_3)$ depend on the masses of the individual fermions and the kinetic-balance condition expressions simplify significantly if all three fermions have equal masses.

From Eq. (6.70), we recognize that the explicitly correlated kinetic-balance condition for three particles features the momentum operator to the power of seven, which is most unfortunate from a computational point of view. However, we also see that the power of the momentum operators decreases with increasing orders of c. The leading terms with respect to c are again the one-fermion kinetic-balance terms and ensure the non-relativistic limit.

Also the two-fermion kinetic-balance condition can be expanded in terms of the type in Eq. (6.70) as

$$d_i^{(2)}(c) = k_i(m_1, m_2) \times c^{(4-u-v)} (\boldsymbol{\sigma}_1 \cdot \boldsymbol{p}_1)^u (\boldsymbol{\sigma}_2 \cdot \boldsymbol{p}_2)^v \ . \tag{6.74}$$

The multiplicative prefactors $k_i(m_1, m_2)$ again depend on the masses of the two fermions. The positive semi-definite exponents u and v obey the constraints $0 \leq (u + v) \leq 3$.

Comparing the results for two and three fermions listed in Eqs. (6.70) and (6.74), we can conclude that the power of c and the power of the $\boldsymbol{\sigma}_i \cdot \boldsymbol{p}_i$ operators behaves as follows for the different terms in the exact solution

$$d_i^{(N)}(c) = k_i(m_1, \ldots, m_N) \times c^{(2N-u)} \prod_{j=1}^{N} (\boldsymbol{\sigma}_j \cdot \boldsymbol{p}_j)^{u_j} \ , \tag{6.75}$$

where

$$0 \leq u = \sum_{j=1}^{N} u_j \leq 2N + 1 \ , \tag{6.76}$$

and where the multiplicative prefactors $k_i(m_1, \ldots, m_N)$ depend on the masses of the different fermions and all exponents are positive semi-definite. In contrast to the approach by Pestka and co-workers [18,71,72], it is now possible to systematically formulate approximate forms for the explicitly correlated kinetic-balance conditions with respect to the order in c, in order to reduce the computational cost associated with high powers of the $\boldsymbol{\sigma}_i \cdot \boldsymbol{p}_i$ operator. As an example, we present the approximate kinetic-balance condition for an electronic system of three particles where only the first few leading terms of

c are included:

$$
\begin{aligned}
|lll\rangle &= \left(48c^6 - 14((\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)^2 + (\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)^2 + (\boldsymbol{\sigma}_3\cdot\boldsymbol{p}_3)^2)c^4\right)\boldsymbol{\Theta} && (6.77)\\
|lls\rangle &= (\boldsymbol{\sigma}_3\cdot\boldsymbol{p}_3)\left((-(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)^2 - (\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)^2 - 7(\boldsymbol{\sigma}_3\cdot\boldsymbol{p}_3)^2)c^3 + 24c^5\right)\boldsymbol{\Theta} && (6.78)\\
|lsl\rangle &= (\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)\left((-(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)^2 - 7(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)^2 - (\boldsymbol{\sigma}_3\cdot\boldsymbol{p}_3)^2)c^3 + 24c^5\right)\boldsymbol{\Theta} && (6.79)\\
|lss\rangle &= 12(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)(\boldsymbol{\sigma}_3\cdot\boldsymbol{p}_3)c^4\boldsymbol{\Theta} && (6.80)\\
|sll\rangle &= (\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)\left((-7(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)^2 - (\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)^2 - (\boldsymbol{\sigma}_3\cdot\boldsymbol{p}_3)^2)c^3 + 24c^5\right)\boldsymbol{\Theta} && (6.81)\\
|sls\rangle &= 12(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)(\boldsymbol{\sigma}_3\cdot\boldsymbol{p}_3)c^4\boldsymbol{\Theta} && (6.82)\\
|ssl\rangle &= 12(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)c^4\boldsymbol{\Theta} && (6.83)\\
|sss\rangle &= -6(\boldsymbol{\sigma}_1\cdot\boldsymbol{p}_1)(\boldsymbol{\sigma}_2\cdot\boldsymbol{p}_2)(\boldsymbol{\sigma}_3\cdot\boldsymbol{p}_3)c^3\boldsymbol{\Theta}\ . && (6.84)
\end{aligned}
$$

Here, we have used the definitions in Eqs. (6.71) – (6.73) for the $\boldsymbol{\sigma}_i\cdot\boldsymbol{p}_i$ operators. $\boldsymbol{\Theta}(\boldsymbol{r})$ with $\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3)^{\mathrm{T}}$ is the non-relativistic limit of $\boldsymbol{\Psi}(\boldsymbol{r})$. We see that the lowest order of c that we need to consider is 3 due to the $|sss\rangle$ component. Eqs. (6.77) – (6.84) can be considered a minimal explicitly correlated kinetic-balance condition for an electronic three-fermion system.

6.7 Basis-Set Expansion and Numerical Results

Until now, we have only considered state functions. In practice, the state function is approximated in terms of a basis set expansion

$$\boldsymbol{\Psi}(\boldsymbol{r}) = \sum_i c_i \boldsymbol{\Phi}_i(\boldsymbol{r}) \tag{6.85}$$

where c_i are the expansion coefficients and $\boldsymbol{\Phi}_i(\boldsymbol{r})$ are the basis functions used in the expansion.

The one-fermion kinetic-balance condition can be imposed by a transformation [48] of the basis functions,

$$\boldsymbol{U}^{(1)}_{\mathrm{KB}} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \dfrac{\boldsymbol{\sigma}\cdot\boldsymbol{p}}{p} \end{bmatrix} \tag{6.86}$$

where $p = |\boldsymbol{p}|$. The model spaces for the large and the small components are generated in terms of this transformation and the advantage of this form of the kinetic-balance condition is that the large-component and small-component model spaces are normalized. This is equivalent to the transformation of the state function in terms of the transformation matrix in Eq. (6.86). Yet, it is also possible to transform the Dirac Hamiltonian and then form identical model spaces for the large and small components. The transformed Dirac

Hamiltonian is the so-called "modified Dirac Hamiltonian" and is the basis of exact decoupling methods [48].

A similar transformation can be formulated for the explicitly correlated kinetic-balance condition as

$$\boldsymbol{U}_{\mathrm{KB}}^{(2)} = \begin{bmatrix} \dfrac{U_{ll}}{|U_{ll}|} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \dfrac{U_{ls}}{|U_{ls}|} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dfrac{U_{sl}}{|U_{sl}|} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dfrac{U_{ss}}{|U_{ss}|} \end{bmatrix} \tag{6.87}$$

in the notation introduced in Eqs. (6.56) – (6.59). The N-fermion trial wave function is then expressed in terms of the transformation as

$$\boldsymbol{\Psi}(\boldsymbol{r}) = \sum_i c_i \left(\boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}_i(\boldsymbol{r}) \right) \ . \tag{6.88}$$

6.7.1 Numerical Results

As an example, we present numerical results for a simple two-electron system: the helium atom treated as two electrons in a central potential in the BO approximation. Since we consider only the ground state, we can generate a trial wave function according to Eq. (6.88) using ECGs [2, 3, 27, 29–31, 34–36, 79–81, 217, 249–257],

$$\boldsymbol{\Theta}(\boldsymbol{r}_1, \boldsymbol{r}_2) = \exp\left(-\frac{1}{2} \boldsymbol{r}^{\mathrm{T}} \left(\boldsymbol{A} \otimes \mathbf{1}_3 \right) \boldsymbol{r} \right) \ , \tag{6.89}$$

where $\boldsymbol{A}$ is a 2×2 positive definite matrix and $\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2)^{\mathrm{T}}$ collects the Cartesian coordinates of the two fermions. Insertion of the trial wave function of Eq. (6.88) into the Rayleigh coefficient

$$E[\boldsymbol{H}_{DTS}^{(2)}, \boldsymbol{\Theta}(\boldsymbol{r})] = \frac{\left\langle \boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}(\boldsymbol{r}) \right| \boldsymbol{H}_{DTS}^{(2)} \left| \boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}(\boldsymbol{r}) \right\rangle}{\left\langle \boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}(\boldsymbol{r}) \right| \left. \boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}(\boldsymbol{r}) \right\rangle} \tag{6.90}$$

and minimization with respect to the expansion coefficients c_i yields the generalized eigenproblem

$$\boldsymbol{HC} = \boldsymbol{ESC} \ . \tag{6.91}$$

The Hamiltonian and overlap matrices are defined as

$$\boldsymbol{H}_{I,J} = \left\langle \boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}_I(\boldsymbol{r}) \right| \boldsymbol{H}_{DTS}^{(2)} \left| \boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}_J(\boldsymbol{r}) \right\rangle \tag{6.92}$$

and

$$\boldsymbol{S}_{I,J} = \left\langle \boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}_I(\boldsymbol{r}) \left| \boldsymbol{U}_{\mathrm{KB}}^{(N)} \boldsymbol{\Theta}_J(\boldsymbol{r}) \right\rangle \right. \tag{6.93}$$

for n basis functions, respectively. They are of dimension $n \times n$. $\boldsymbol{C}$ is a $n \times n$ matrix containing the expansions coefficients c_i and $\boldsymbol{E}$ is a $n \times n$ diagonal-matrix with the energies as its entries.

Table 6.1 lists the energies we obtained for the ground state of the helium atom for various sizes of the parameter sets together with the non-relativistic limit of each parameter set. We recognize that both the relativistic energy and the non-relativistic energy continuously approach the ground state energy of the reference, forming an upper limit to the actual energy.

Table 6.1: Energies of the He$= \{e^-, e^-\}$ atom for different parameter sets. ΔE_{R} and ΔE_{NR} are the differences between the calculated relativistic and non-relativistic energies in comparison to the reference values, respectively. With increasing size of the parameter set, the energy gradually converges towards the reference values in a variationally stable fashion.

n	E_{R} $[E_h]$	ΔE_{R} $[E_h]$	E_{NR} $[E_h]$	ΔE_{NR} $[E_h]$
50	-2.90377913	0.00007771	-2.90364752	0.00007686
100	-2.90385291	0.00000393	-2.90372249	0.00000189
500	-2.90385512	0.00000172	-2.90372403	0.00000035
1000	-2.90385650	0.00000034	-2.90372431	0.00000007
Ref. [72]	-2.90385684	Ref. [258]	-2.90372438	

7

Dirac–Coulomb Fine-Structure of Hydrogen-Like Ions

In this chapter, we extend our non-relativistic pre-BO framework [29, 30, 81, 217] to a relativistic first-quantized formulation employing the Dirac Hamiltonian [55, 259], which is the basis of most of relativistic quantum chemistry [7–13]. Here, only the matter field is quantized, while the radiation field is described classically.

Despite the popularity and importance of the Dirac Hamiltonian in relativistic quantum chemistry, there exist few explicitly correlated methods due to variational problems caused by the unboundedness of the energy spectrum of the Dirac Hamiltonian. For so-called four-component methods, variational stability is ensured through the kinetic-balance condition [15]. For orbital-based methods, the one-fermion kinetic-balance is a reliable means of avoiding variational collapse. Pestka and co-workers have presented an approximate form of explicitly correlated kinetic-balance without explaining the means for generalization for more than two particles [58, 72, 182, 189, 192, 193]. The exact explicitly correlated variant was presented in chapter 6.

We treat all particles on equal footing and therefore do not invoke the BO approximation. Hence, our framework represents a pre-BO theory (note that the term ”non-Born–Oppenheimer” may refer to various theoretical improvements that go beyond the BO approximation while we want to stress by ”pre-BO” that the ”ubiquitous” BO approximation is not applied at all). As a basic problem is related to the overall motion of the system we have to eliminate translational effects in order to be able to formulate a translationally invariant theory.

7.1 Relativistic Trial Wave Function

The trial wave function employed in this work is partitioned according to Eq. (6.22) with the components labeled according to Eq. (6.33) and approximates the eigenfunction $\boldsymbol{\Psi}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ of the Dirac Hamiltonian in Eq. (3.7) as a linear combination of explicitly correlated basis functions $\boldsymbol{\Phi}_i(\boldsymbol{r}_1, \boldsymbol{r}_2)$

$$\boldsymbol{\Psi}(\boldsymbol{r}_1, \boldsymbol{r}_2) = \sum_i \sum_{\lambda \in \Lambda} c_i^\lambda \boldsymbol{\Phi}_i^\lambda(\boldsymbol{r}_1, \boldsymbol{r}_2) \ . \tag{7.1}$$

Here, $\boldsymbol{\Phi}_i^\lambda(\boldsymbol{r}_1, \boldsymbol{r}_2)$ are the components of the individual basis functions where $\lambda \in \Lambda = \{ll, ls, sl, ss\}$ indicates the component. Each basis function is related to its non-relativistic limit $\boldsymbol{\Theta}_i(\boldsymbol{r}_1, \boldsymbol{r}_2)$ as [260]

$$\boldsymbol{\Phi}_i(\boldsymbol{r}_1, \boldsymbol{r}_2) = \boldsymbol{U}_{\mathrm{KB}}^{(2)} \boldsymbol{\Theta}_i(\boldsymbol{r}_1, \boldsymbol{r}_2) \ . \tag{7.2}$$

The trial wave function, is therefore generated from its non-relativistic limit and the linear expansion in Eq. (7.1) can then be formulated as

$$\boldsymbol{\Psi}(\boldsymbol{r}_1, \boldsymbol{r}_2) = \sum_i \sum_{\lambda \in \Lambda} c_i^\lambda \left(\boldsymbol{U}_{\mathrm{KB}}^{(2)} \boldsymbol{\Theta}_i(\boldsymbol{r}_1, \boldsymbol{r}_2) \right)^\lambda \ . \tag{7.3}$$

We now need to choose proper basis functions for the $\boldsymbol{\Theta}_i(\boldsymbol{r}_1, \boldsymbol{r}_2)$ part of the basis functions $\boldsymbol{\Phi}_i(\boldsymbol{r}_1, \boldsymbol{r}_2)$. Since pre-BO theories are well established for the non-relativistic theory, we follow recipes given in previous work [29, 30, 81, 217] as a guide for choosing efficient basis functions.

The spatial part of $\boldsymbol{\Theta}_i(\boldsymbol{r}_1, \boldsymbol{r}_2)$ will be represented by ECGs in the original LFCC

$$G_i(\boldsymbol{r}; \boldsymbol{A}_i) = \exp\left(-\frac{1}{2} \boldsymbol{r}^{\mathrm{T}} (\boldsymbol{A}_i \otimes \boldsymbol{1}_3) \boldsymbol{r} \right) \tag{7.4}$$

where $\boldsymbol{A}_i$ is a positive definite 2×2 matrix and $\boldsymbol{r}^{\mathrm{T}} = (\boldsymbol{r}_1^{\mathrm{T}}, \boldsymbol{r}_2^{\mathrm{T}})$ collects the Cartesian coordinates of the two fermions. ECGs were originally introduced by Boys [79] and Singer [80] and have been successfully used in our previous non-relativistic work [29, 30, 81, 217]. ECGs are also commonly used by Adamowicz and co-workers for their non-relativistic work [3, 27, 31, 34, 249–257] and by Suzuki and Varga [2, 35, 36].

In a series of papers, Marsch has derived an analytical solution for the two-fermion Dirac Hamiltonian [261–264]. In a first step, angular and radial degrees of freedom are separated. This leads to the total angular momentum operator

$$\hat{\boldsymbol{J}} = \hat{\boldsymbol{L}} + \hat{\boldsymbol{S}} \tag{7.5}$$

where $\hat{\boldsymbol{L}}$ is the total spatial angular momentum operator and $\hat{\boldsymbol{S}}$ is the total spin operator. Its square

$$\hat{\boldsymbol{J}}^2 = \hat{\boldsymbol{L}}^2 + 2\hat{\boldsymbol{L}} \cdot \hat{\boldsymbol{S}} + \hat{\boldsymbol{S}}^2 \tag{7.6}$$

and projection onto the z-axis

$$\hat{\boldsymbol{J}}_z = \hat{\boldsymbol{L}}_z + \hat{\boldsymbol{S}}_z \tag{7.7}$$

commute with the two-fermion Dirac Hamiltonian and the angular part of $\boldsymbol{\Theta}_i(\boldsymbol{r}_1, \boldsymbol{r}_2)$ is constructed from the GVR [2,35,36]

$$\mathcal{Y}_L^{M_L}(\boldsymbol{v}) = |\boldsymbol{v}|^{2K+L} Y_{M_L}^L(\hat{\boldsymbol{v}}) \ , \tag{7.8}$$

with the global vector

$$\boldsymbol{v} = (\boldsymbol{u} \otimes \mathbf{1}_3)\boldsymbol{r} \ , \tag{7.9}$$

and the strictly non-negative integer K, as the eigenfunction of operators $\boldsymbol{J}^2$ and $\boldsymbol{J}_z$. In this work, we focus on the case where $K = 0$. The GVR is an eigenfunction of the total spatial angular momentum operators L^2 and L_z in non-relativistic theory describing the spatial angular momentum state in terms of the quantum numbers L and M_L. The GVR has been previously used in our non-relativistic work [29,30,81,217]. For the spin-triplet state, we find two possible representations. The first one, where $k = J = L + 1$, is

$$\boldsymbol{\theta}_L^{M_L}(\boldsymbol{v}) = \sqrt{\frac{(L+1)^2 - M_L^2}{2(L+1)(2L+1)}} \begin{bmatrix} \sqrt{\frac{L+M_L}{L-M_L+1}}\mathcal{Y}_L^{M_L-1}(\boldsymbol{v}) \\ \mathcal{Y}_L^{M_L}(\boldsymbol{v}) \\ \mathcal{Y}_L^{M_L}(\boldsymbol{v}) \\ \sqrt{\frac{L-M_L}{L-M_L+1}}\mathcal{Y}_L^{M_L+1}(\boldsymbol{v}) \end{bmatrix} \ , \tag{7.10}$$

and the second one, where $k = -l$ and $j = l - 1$, is

$$\boldsymbol{\theta}_L^{M_L}(\boldsymbol{v}) = \sqrt{\frac{L^2 - M_L^2}{2L(2L+1)}} \begin{bmatrix} -\sqrt{\frac{L-M_L+1}{L+M_L}}\mathcal{Y}_L^{M_L-1}(\boldsymbol{v}) \\ \mathcal{Y}_L^{M_L}(\boldsymbol{v}) \\ \mathcal{Y}_L^{M_L}(\boldsymbol{v}) \\ -\sqrt{\frac{L+M_L+1}{L-M_L}}\mathcal{Y}_L^{M_L+1}(\boldsymbol{v}) \end{bmatrix} \ . \tag{7.11}$$

Here, we have introduced the auxiliary quantum number k.

The well known fine-structure of the hydrogen atom only shows up in the spin-triplet state. The spin-singlet state lacks the spin-orbit splitting. We do only consider triplet hydrogen in this work.

The ECGs and the total angular momentum function formulated in terms of the GVR can then be combined to form

$$\begin{aligned}\boldsymbol{\Theta}_i(\boldsymbol{r}_1,\boldsymbol{r}_2) &= \boldsymbol{\Theta}^J_{M_J\ i}(\boldsymbol{r}_1,\boldsymbol{r}_2;\boldsymbol{u}_i,\boldsymbol{A}_i)\\ &= \sqrt{\frac{(L+1)^2-M_L^2}{2(L+1)(2L+1)}}\begin{bmatrix}\sqrt{\frac{L+M_L}{L-M_L+1}}\mathcal{Y}_L^{M_L-1}(\boldsymbol{v}_i)\\ \mathcal{Y}_L^{M_L}(\boldsymbol{v}_i)\\ \mathcal{Y}_L^{M_L}(\boldsymbol{v}_i)\\ \sqrt{\frac{L-M_L}{L-M_L+1}}\mathcal{Y}_L^{M_L+1}(\boldsymbol{v}_i)\end{bmatrix}\exp\left(-\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{A}_i\otimes\mathbf{1}_3)\boldsymbol{r}\right)\end{aligned} \tag{7.12}$$

for states with $J = L+1$. The matrix $\boldsymbol{A}_i$ and the global vector $\boldsymbol{u}_i$ are the two non-linear variational parameters of basis function i. Their structure is discussed in chapter 4.

The linear factors c_i^λ in Eq. (7.3) are determined such that they minimize the Rayleigh quotient in Eq. (3.29). This leads to the generalized eigenproblem

$$\boldsymbol{HC} = \boldsymbol{ESC} \tag{7.13}$$

which is solved for the diagonal $(n\times 2^N)\times(n\times 2^N)$ matrix $\boldsymbol{E}$ containing the energies of the different eigenstates and the $(n\times 2^N)\times(n\times 2^N)$ matrix $\boldsymbol{C}$ which contains the expansion coefficients from Eq. (7.1). $\boldsymbol{H}$ and $\boldsymbol{S}$ are the matrix representations of the Hamiltonian and the overlap with the elements defined as

$$\boldsymbol{H}_{(I\times n+\lambda_I),(J\times n+\lambda_J)} = \left\langle (\boldsymbol{U}^{(N)}_{\mathrm{KB}}\boldsymbol{\Theta}_I(\boldsymbol{r}))^{\lambda_I}\middle|\,\boldsymbol{H}^{(2)}_{DTS}\,\middle|(\boldsymbol{U}^{(N)}_{\mathrm{KB}}\boldsymbol{\Theta}_J(\boldsymbol{r}))^{\lambda_J}\right\rangle \tag{7.14}$$

and

$$\boldsymbol{S}_{(I\times n+\lambda_I),(J\times n+\lambda_J)} = \left\langle (\boldsymbol{U}^{(N)}_{\mathrm{KB}}\boldsymbol{\Theta}_I(\boldsymbol{r}))^{\lambda_I}\,\middle|(\boldsymbol{U}^{(N)}_{\mathrm{KB}}\boldsymbol{\Theta}_J(\boldsymbol{r}))^{\lambda_J}\right\rangle\ , \tag{7.15}$$

respectively. The two matrices are of dimension $(n\times 2^N)\times(n\times 2^N)$. We use the indices I, J, λ_I and λ_J which run from zero in order to obtain simpler relations.

The generalized eigenvalue problem corresponding to the matrix representation of the Hamiltonian is solved using the standard linear algebra library routines of LAPACK (Version 3.2.1) [265] through the Armadillo framework (Version 3.4.0) [266].

7.2 Integrals over Cartesian Explicitly Correlated Gaussians

The kinetic-balance condition presented in Eqs. (6.56) – (6.59) requires the action of the $\boldsymbol{\sigma}\cdot\boldsymbol{p}$ operator on the function $\boldsymbol{\Theta}^J_{M_J\ i}(\boldsymbol{r}_1,\boldsymbol{r}_2;\boldsymbol{u}_i,\boldsymbol{A}_i)$ defined in

Eq. (7.12). Unfortunately, the basis function does not maintain its general form due to the GVR. We therefore have to recast the GVR in a form which is invariant under the action of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator. The simplest form are explicitly correlated Gaussians with Cartesian polynomial prefactors (CECGs):

$$G(\boldsymbol{r}; \boldsymbol{A})[\boldsymbol{l}] = \prod_{n=1}^{N} \prod_{\mu \in \{x,y,z\}} \boldsymbol{r}_{n\mu}^{\boldsymbol{l}_{n\mu}} G(\boldsymbol{r}; \boldsymbol{A}) \tag{7.16}$$

where $\boldsymbol{l}$ is a vector containing the exponents for the different factors in the polynomial and where we have omitted the basis function index for the sake of brevity. For $K = 0$, the GVR can be expressed as a linear combination of Cartesian polynomials [2] according to

$$\begin{aligned} \mathcal{Y}_L^{M_L}(\boldsymbol{v}) = \sqrt{\frac{2L+1}{4\pi}(L+M_L)!(L-M_L)!} \sum_{p=\max(0,M_L)}^{\left\lfloor \frac{L-M_L}{2} \right\rfloor} \frac{2^{M_L-2p} \boldsymbol{v}_z^{L+M_L-2p}}{(L+M_L-2p)!} \\ \times \sum_{a=0}^{p} \sum_{b=0}^{p-M_L} (-1)^{M_L+b} \frac{\boldsymbol{v}_x^{a+b} (i\boldsymbol{v}_y)^{2p-a-b-M_L}}{a!b!(p-a)!(p-M_L-b)!} \ . \end{aligned} \tag{7.17}$$

Our ansatz for $\boldsymbol{\Theta}^{J}_{M_J\ i}(\boldsymbol{r}_1, \boldsymbol{r}_2; \boldsymbol{u}_i, \boldsymbol{A}_i)$ in Eq. (7.12) can then be written as a linear combination of CECGs,

$$\boldsymbol{\Theta}^{J}_{M_J\ i}(\boldsymbol{r}_1, \boldsymbol{r}_2; \boldsymbol{u}_i, \boldsymbol{A}_i) = \sum_{j=1} c_j(\boldsymbol{u}_i) G_j(\boldsymbol{r}; \boldsymbol{A})[\boldsymbol{l}_j] \ , \tag{7.18}$$

where the expansion coefficients $c_j(\boldsymbol{u}_i)$ and the polynomial exponents $\boldsymbol{l}_j$ are determined according to Eq. (7.17).

Expressions for integrals over CECGs were presented by Boys [79], Singer [80], Kozlowski and Adamowicz [254], and Cencek and Rychlewski [213]. As an alternative, Saito and Suzuki [267] have presented recursion relations for integrals over CECGs involving the Fourier kernel. This approach is far less involved, especially for CECGs centered at the origin. The Fourier kernel limits the applicability of this approach when integrals for inverse-law potentials

$$V_C(\lambda) = |\boldsymbol{e}^T r|^{-\lambda} \tag{7.19}$$

are involved. $\boldsymbol{e}$ is an N-vector defining the linear combination of one-particle coordinates. For a single particle j, $\boldsymbol{e}$ has the elements $\boldsymbol{e}_i = \delta_{ij}$ and the potential becomes

$$V_C(\lambda) = |\boldsymbol{r}_j|^{-\lambda} \tag{7.20}$$

while for a potential depending on the distance of two particles p and q, $\boldsymbol{e}$ becomes $\boldsymbol{e}_i = \delta_{ip} - \delta_{iq}$. Since there will be no Fourier transform for Eq. (7.19) if $\lambda > 1$, it can only be applied to Coulomb-type interactions but not for Breit-type interactions which contain terms with $\lambda = 3$. An alternative to the Fourier kernel is the modified Laplace kernel. The integral transform of Eq. (7.19) then reads

$$|\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r}|^{-\lambda} = \int_0^\infty du\ \frac{2u^{\lambda-1}}{\Gamma(\lambda/2)} \exp\left(-u^2(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r})^2\right)\ . \tag{7.21}$$

This Laplace kernel has been exploited by Obara and Saika [268, 269] for the four-center electron repulsion integrals and also by Shiozaki for the Breit interaction [270]. Their work, however, is based on orbitals, not ECGs. We therefore have to adjust the derivation for origin-centered CECGs.

The integrals involve the product of two origin-centered CECGs multiplied with the modified Laplace kernel. The product of the two CECGs conveniently leads to a new CECG also centered at the origin

$$\begin{aligned} G(\boldsymbol{r})[\boldsymbol{l}] \equiv G[\boldsymbol{l}] &= G(\boldsymbol{r};\boldsymbol{A}_I)[\boldsymbol{l}_I] G(\boldsymbol{r};\boldsymbol{A}_J)[\boldsymbol{l}_J] \exp\left(-u^2(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r})^2\right) \\ &= \exp\left(-\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\left(\boldsymbol{A}\otimes\mathbf{1}_3\right)\boldsymbol{r}\right) \prod_{n=1}^{N} \prod_{\mu\in\{x,y,z\}} \boldsymbol{r}_{n\mu}^{\boldsymbol{l}_{n\mu}} \exp\left(-u^2(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r})^2\right) \\ &= G(\boldsymbol{r};\boldsymbol{A})[\boldsymbol{l}] \exp\left(-u^2(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r})^2\right) \end{aligned} \tag{7.22}$$

where

$$\boldsymbol{A} = \boldsymbol{A}_I + \boldsymbol{A}_J \tag{7.23}$$

and

$$\boldsymbol{l} = \boldsymbol{l}_I + \boldsymbol{l}_J\ . \tag{7.24}$$

First, the primitive integral

$$[\boldsymbol{l}]_N = \int_{-\infty}^{\infty} d\boldsymbol{r}\ \ G[\boldsymbol{l}] \tag{7.25}$$

is used to derive the basic recurrence relation. All further integral expressions can then be obtained through reduction operators acting on the primitive integral and its recurrence relation. The recurrence relation is obtained from the derivative of Eq. (7.22) with respect to $\boldsymbol{r}_{i\mu}$

$$\frac{\partial}{\partial \boldsymbol{r}_{i\mu}} G[\boldsymbol{l}] = \boldsymbol{l}_{i\mu} G[\boldsymbol{l}_i - \mathbf{1}_\mu] - \sum_{m=1}^{N} \left[\boldsymbol{A} - 2u^2\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right]_{im} G[\boldsymbol{l}_i + \mathbf{1}_\mu]\ . \tag{7.26}$$

Note, that we follow the notation of Ref. [267] and only print those exponents $\boldsymbol{l}_i$ which change in the manipulation for the sake of brevity. Integrating both sides over $\boldsymbol{r}$ leads to

$$0 = \boldsymbol{l}_{i\mu}[\boldsymbol{l}_i - \boldsymbol{1}_\mu]_N - \sum_{m=1}^{N} \left[\boldsymbol{A} - 2u^2 \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right]_{im} [\boldsymbol{l}_i + \boldsymbol{1}_\mu]_N \tag{7.27}$$

since

$$\int_{-\infty}^{\infty} d\boldsymbol{r} \; \frac{\partial}{\partial \boldsymbol{r}_{i\mu}} G[\boldsymbol{l}] = 0 \; . \tag{7.28}$$

We can now from a vector equation according to Eq. (7.27)

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{l}_{1\mu}[\boldsymbol{l}_1 - \boldsymbol{1}_\mu]_N \\ \boldsymbol{l}_{2\mu}[\boldsymbol{l}_2 - \boldsymbol{1}_\mu]_N \\ \vdots \\ \boldsymbol{l}_{N\mu}[\boldsymbol{l}_N - \boldsymbol{1}_\mu]_N \end{bmatrix} - \left[\boldsymbol{A} - 2u^2 \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\right] \begin{bmatrix} [\boldsymbol{l}_1 + \boldsymbol{1}_\mu]_N \\ [\boldsymbol{l}_2 + \boldsymbol{1}_\mu]_N \\ \vdots \\ [\boldsymbol{l}_N + \boldsymbol{1}_\mu]_N \end{bmatrix} \tag{7.29}$$

which can be rearranged into

$$\begin{bmatrix} [\boldsymbol{l}_1 + \boldsymbol{1}_\mu]_N \\ [\boldsymbol{l}_2 + \boldsymbol{1}_\mu]_N \\ \vdots \\ [\boldsymbol{l}_N + \boldsymbol{1}_\mu]_N \end{bmatrix} = \left[\boldsymbol{A} - 2u^2 \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right]^{-1} \begin{bmatrix} \boldsymbol{l}_{1\mu}[\boldsymbol{l}_1 - \boldsymbol{1}_\mu]_N \\ \boldsymbol{l}_{2\mu}[\boldsymbol{l}_2 - \boldsymbol{1}_\mu]_N \\ \vdots \\ \boldsymbol{l}_{N\mu}[\boldsymbol{l}_N - \boldsymbol{1}_\mu]_N \end{bmatrix} \; . \tag{7.30}$$

The inverse of $[\boldsymbol{A} - 2u^2 \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}]$ is problematic in its current form, since it depends directly on u over which we will later integrate. The factor $2u^2 \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}$ is a rank-one correction to the inverse of $\boldsymbol{A}$ and we can use the Sherman–Morrison formula [271–273] to obtain a more convenient form

$$\begin{aligned} \left[\boldsymbol{A} - 2u^2 \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right]^{-1} &= \boldsymbol{A}^{-1} - \frac{2u^2}{1 + 2u^2 \boldsymbol{e}^{\mathrm{T}} \boldsymbol{A}^{-1} \boldsymbol{e}} \boldsymbol{A}^{-1} \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}} \boldsymbol{A}^{-1} \\ &= \boldsymbol{A}^{-1} - 2\rho \frac{u^2}{\rho + u^2} \boldsymbol{Z} \end{aligned} \tag{7.31}$$

where we have introduced the factors

$$1/\rho = 2\boldsymbol{e}^{\mathrm{T}} \boldsymbol{A}^{-1} \boldsymbol{e} \tag{7.32}$$

and

$$\boldsymbol{Z} = \boldsymbol{A}^{-1} \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}} \boldsymbol{A}^{-1} \; . \tag{7.33}$$

Based on Eqs. (7.30) and (7.31) we find the primitive recurrence relation

$$[\boldsymbol{l}_i + \mathbf{1}_\mu]_N = \sum_{m=1}^{N} \left(\boldsymbol{A}_{mi}^{-1} - 2\rho \frac{u^2}{\rho + u^2} \boldsymbol{Z}_{mi} \right) \boldsymbol{l}_{m\mu} [\boldsymbol{l}_m - \mathbf{1}_\mu]_N \tag{7.34}$$

with the initial integral

$$\begin{aligned} [\mathbf{0}]_N &= \int_{-\infty}^{\infty} d\boldsymbol{r} \exp\left(-\frac{1}{2} \boldsymbol{r}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{r} - u^2 (\boldsymbol{e}^{\mathrm{T}} \boldsymbol{r})^2 \right) \\ &= \left(\frac{(2\pi)^N}{\det\left(\boldsymbol{A} + 2u^2 \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)} \right)^{3/2} . \end{aligned} \tag{7.35}$$

Again, we would like to separate u in this expression since we will integrate over it later. Using the matrix determinant lemma [274] we find

$$\det\left(\boldsymbol{A} + 2u^2 \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right) = \det\left(\boldsymbol{A}\right) \left(\mathbf{1} + 2u^2 \boldsymbol{e}^{\mathrm{T}} \boldsymbol{A}^{-1} \boldsymbol{e}\right) = \det\left(\boldsymbol{A}\right) \left(1 + u^2/\rho\right) \tag{7.36}$$

where we have again exploited that $2\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}$ has rank one. The primitive initial integral then becomes

$$[\mathbf{0}]_N = \left(\frac{(2\pi)^N}{\det\left(\boldsymbol{A}\right)\left(1 + u^2/\rho\right)} \right)^{3/2} . \tag{7.37}$$

7.2.1 Overlap and (Non-)Relativistic Kinetic-Energy Integral

Acting with the reduction operator

$$\mathcal{R}_S \equiv \lim_{u \to 0} \tag{7.38}$$

on the primitive recursion relation in Eq. (7.34) and the primitive initial integral in Eq. (7.37) leads to the overlap integrals of CECGs $S[\boldsymbol{l}]_N$:

$$S[\boldsymbol{l}_i + \mathbf{1}_\mu]_N = \sum_{m=1}^{N} \boldsymbol{A}_{mi}^{-1} \boldsymbol{l}_{m\mu} S[\boldsymbol{l}_m - \mathbf{1}_\mu]_N \tag{7.39}$$

$$S[\mathbf{0}]_N = \left(\frac{(2\pi)^N}{\det\left(\boldsymbol{A}\right)} \right)^{3/2} . \tag{7.40}$$

We can avoid the formulation of reduction operators for both the non-relativistic kinetic-energy operator and the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator by exploiting the fact the CECGs maintain their general form when these operators act on them and then use the expressions for the overlap integral after the transformation.

7.2.2 Inverse Law Potential Integral

The general reduction operator for inverse law potentials described in Eq. (7.19) is

$$\mathcal{R}_V[m;\lambda] \equiv 2\int_0^\infty du \left(\frac{u^2}{\rho+u^2}\right)^m \frac{u^{\lambda-1}}{\Gamma(\lambda/2)} \tag{7.41}$$

where m is an auxiliary index. Reduction to the inverse law potential integral

$$V[\boldsymbol{l};m;\lambda] = \mathcal{R}_V[m,\lambda][\boldsymbol{l}]_N \tag{7.42}$$

leads to the recursion relation

$$\begin{aligned} V[\boldsymbol{l}_i+\mathbf{1}_\mu;m;\lambda]_N = &\sum_{m=1}^N \boldsymbol{A}^{-1}_{mi}\boldsymbol{l}_{m\mu} V[\boldsymbol{l}_m-\mathbf{1}_\mu;m;\lambda]_N \\ &- 2\rho\sum_{m=1}^N \boldsymbol{Z}_{mi}\boldsymbol{l}_{m\mu} V[\boldsymbol{l}_m-\mathbf{1}_\mu;m+1;\lambda]_N \end{aligned} \tag{7.43}$$

with the initial integral

$$V[\mathbf{0};m;\lambda]_N = 2\int_0^\infty du \left(\frac{u^2}{\rho+u^2}\right)^m \frac{u^{\lambda-1}}{\Gamma(\lambda/2)} \left(\frac{(2\pi)^N}{\det(\boldsymbol{A})\,(\mathbf{1}+u^2/\rho)}\right)^{3/2} \tag{7.44}$$

$$= \frac{2S[\mathbf{0}]}{\Gamma(\lambda/2)}\int_0^\infty du \left(\frac{u^2}{\rho+u^2}\right)^m u^{\lambda-1}\left(1+u^2/\rho\right)^{-3/2} \ . \tag{7.45}$$

The integral over u can be simplified by introducing the substitution

$$t = \frac{u^2}{\rho+u^2} \qquad \Longrightarrow \qquad u^2 = \rho\frac{t}{1-t} \tag{7.46}$$

so that

$$du = \frac{1}{2}\left(\rho\frac{t}{1-t}\right)^{-1/2}(1-t)^{-2}dt \ . \tag{7.47}$$

The initial integral then takes a new form as

$$\begin{aligned} V[\mathbf{0};m;\lambda]_N = &\frac{2S[\mathbf{0}]}{\Gamma(\lambda/2)}\rho^{(\lambda-1)/2} \\ &\times \int_0^1 dt\ t^m \left(\frac{t}{1-t}\right)^{(\lambda-1)/2}\left(1+\frac{t}{1-t}\right)^{-3/2}(1-t)^{-2} \ . \end{aligned} \tag{7.48}$$

Partial fraction decomposition allows us to reformulate the term

$$\frac{t}{1-t} = \frac{1}{1-t} - 1 \tag{7.49}$$

which simplifies Eq. (7.48) as

$$V[\mathbf{0};m;\lambda]_N = \frac{2S[\mathbf{0}]}{\Gamma(\lambda/2)}\rho^{\lambda/2}B(m+\lambda/2,1.5-\lambda/2) \tag{7.50}$$

in terms of the beta function

$$B\left(x,y\right) = \int_0^1 dt\ t^{x-1}(1-t)^{y-1}\ . \tag{7.51}$$

The ratio

$$K[m;\lambda] = \frac{B(m+(\lambda-1)/2,3-\lambda/2)}{\Gamma(\lambda/2)} \tag{7.52}$$

depends on two non-negative integers m and λ. Calculations can therefore be performed more efficiently by precalculating $K[m,\lambda]$ for a suitable range of m values for a specific choice of λ, i.e., a specific type of interaction.

The Breit term in Eq. (3.10) is rarely included in relativistic electronic-structure calculations. This is due to the $\boldsymbol{r}^{-3}$ term present in the operator, which leads to expensive two-electron four-center integrals. The results presented by Shiozaki, however, show that it is possible to evaluate this integral at a similar computational cost, as the Coulomb integral. Our derivation is based on the same strategy and, not surprisingly, we find that the computational cost for our integrals is nearly independent of λ.

7.2.3 Translationally Invariant Integrals

To identify contributions from c_A to some function $F(\boldsymbol{A},\boldsymbol{u})$ we use the substitution

$$F(\boldsymbol{A},\boldsymbol{u}) = F(\boldsymbol{U}_x^{\mathrm{T}}\boldsymbol{A}^{(x)}\boldsymbol{U}_x, \boldsymbol{U}_x^{\mathrm{T}}\boldsymbol{u}^{(x)})\ . \tag{7.53}$$

We illustrate this procedure for the inverse of the sum of two matrices $\boldsymbol{A}_I$ and $\boldsymbol{A}_J$ which is important for the integrals in Eqs. (7.39) and (7.43). We start with the substitution and form the inverse

$$(\boldsymbol{A}_I+\boldsymbol{A}_J)^{-1} = \left(\boldsymbol{U}_x^{\mathrm{T}}(\boldsymbol{A}_I^{(x)}+\boldsymbol{A}_J^{(x)})\boldsymbol{U}_x\right)^{-1} = \boldsymbol{U}_x^{-1}(\boldsymbol{A}_I^{(x)}+\boldsymbol{A}_J^{(x)})^{-1}\boldsymbol{U}_x^{-\mathrm{T}}\ . \tag{7.54}$$

Studying the block structure of the general form of the transformation matrix in Eq. (2.6) and $\boldsymbol{A}^{(x)}$

$$\begin{aligned}\boldsymbol{U}_x^{-1}(\boldsymbol{A}_I^{(x)}+\boldsymbol{A}_J^{(x)})^{-1}\boldsymbol{U}_x^{-\mathrm{T}} &= \begin{bmatrix}\boldsymbol{M}' & \boldsymbol{\epsilon}\end{bmatrix}\begin{bmatrix}(\mathcal{A}_I^{(x)}+\mathcal{A}_J^{(x)})^{-1} & \mathbf{0}\\ \mathbf{0} & 1/(2c_A)\end{bmatrix}\begin{bmatrix}\boldsymbol{M}'^{\mathrm{T}}\\ \boldsymbol{\epsilon}^{\mathrm{T}}\end{bmatrix}\\ &= \boldsymbol{M}'(\mathcal{A}_I^{(x)}+\mathcal{A}_J^{(x)})^{-1}\boldsymbol{M}'^{\mathrm{T}} + \frac{1}{2c_A}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}\ .\end{aligned} \tag{7.55}$$

This leads to the tranlationally invariant inverse

$$(\boldsymbol{A}_I + \boldsymbol{A}_J)^{-1}_{TI} = (\boldsymbol{A}_I + \boldsymbol{A}_J)^{-1} - \frac{1}{2c_A}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}} \tag{7.56}$$

where $\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}$ evaluates to a $N \times N$ matrix filled with ones. So we can simply subtract $1/2c_A$ from each element in the inverse in order to obtain its translationally invariant form.

Following this procedure, it becomes clear that the initial integrals in Eqs. (7.39) and (7.43) are free from translational effects as long as c_u is zero and the only contribution of c_A is eliminated by using the translationally invariant inverse in Eq. (7.56).

7.3 Results

We first focus on the results for atomic hydrogen and compare them to the analytical results obtained by Marsch [261]. Note that the analytical expressions for the energy given in Refs. [261–264] are not identical. Unfortunately the reason for these inconsistencies remain unknown [275].

The trial wave functions contained 100 basis functions for all calculations. The non-linear parameters $\boldsymbol{A}_i$ and $\boldsymbol{u}_i$ in Eq. (7.12) are generated using a system adapted random number generator according to Ref. [30] following the variational stochastic sampling method [2,41,42]. The mass of the proton m_{p} was chosen in terms of the electron mass m_{e} as $m_{\mathrm{p}}/m_{\mathrm{e}} = 1836.15267247$ [236]. Table 7.1 lists the non-relativistic energies in the first half and the Dirac-Coulomb results in the second half. We obtain results converged to at least 7 significant digits.

Next, we study the non-relativistic limit for $c \to \infty$ and the BO limit for $m_p \to \infty$. Table 7.2 collects results for large values of c (10^5 a.u.) and m_p (10^{12} a.u.). Clearly, the correct limits are obtained.

Finally, we present the ground-state energies for hydrogen-like ions with nuclear charges between 2 and 10. We have selected isotopes with nuclear spin 1/2. The elements for which no isotope with nuclear spin 1/2 exists were not considered. Table 7.3 presents both the relativistic and non-relativistic ground state energies of seven isotopes. We should note that Marsch [261] did not include the nuclear charge into the derivation. However, following the original derivation, we find that the energy expression can easily be expanded as

$$E(n,k) = (m_1 + m_2)c^2\sqrt{1 + 2\frac{m_1 m_2}{(m_1 + m_2)^2}(e(n,k) - 1)} \tag{7.57}$$

Table 7.1: Results for atomic hydrogen and values from the literature. We used 100 basis functions in each calculation. All results are in Hartree atomic units and relativistic results are printed to the first differing digit compared to the reference value. The non-relativistic parameter sets of non-linear variational parameters was used to generate the relativistic parameter sets.

Non-relativistic Results			Dirac–Coulomb Results		
$(v,L,p)^a$	E_{nr} [$\mathrm{E_h}$]	Analyt. E_{nr} [$\mathrm{E_h}$]^b	$(v,L,k=J,p)^a$	E_{DC} [$\mathrm{E_h}$]	Analyt. E_{DC} [$\mathrm{E_h}$]^c
$(0,0,+1)$	-0.499727839	-0.499727839	$(0,0,1,+1)$	-0.49973446	-0.49973449
$(1,0,+1)$	-0.124931959	-0.124931959	$(1,0,1,+1)$	-0.12493403	-0.12493404
$(2,0,+1)$	-0.055525315	-0.055525315	$(2,0,1,+1)$	-0.055526053	-0.055526054
$(0,1,-1)$	-0.124931959	-0.124931959	$(0,1,2,-1)$	-0.124932374	-0.124932375
$(1,1,-1)$	-0.055525315	-0.055525315	$(1,1,2,-1)$	-0.055525561	-0.055525562

a : v: vibrational quantum number, L: total spatial angular momentum, p: parity, J: total angular momentum, k: auxiliary quantum number.
b: The reference energy of non-relativistic pre-BO hydrogen-like atoms are calculated as $E = -1/2\ m_n m_e/(m_n + m_e)\ Z^2/n^2$ where m_n is the nuclear mass, Z is the nuclear charge number and $n = v + L$ is the principal quantum number.
c : The reference energies are taken from Ref. [261].

Table 7.2: Our calculated (C) non-relativistic limit (E_{nrl}) with $c = 10^5$ a.u. , BO limit (E_{bol}) with $m_p = 10^{12}$ a.u. and the non-relativistic BO limit (E_{nbl}) with $c = 10^5$ a.u. and $m_p = 10^{12}$ a.u. for the hydrogen atom in comparison with the according analytically exact values (A).

	E [$\mathrm{E_h}$]	E_{nrl} [$\mathrm{E_h}$]^a	E_{bol} [$\mathrm{E_h}$]^b	E_{nbl} [$\mathrm{E_h}$] ^{c}
Ground State $(v = 0, L = 0, k = 1, p = +1)$				
C	-0.49973446	-0.499727839	-0.500006656	-0.499999999
A	-0.49973449	-0.499727839	-0.500006656	-0.500000000
First Excited State $(v = 1, L = 0, k = 1, p = +1)$				
C	-0.12493403	-0.124931959	-0.125002080	-0.1250000000
A	-0.12493404	-0.124931959	-0.125002080	-0.1250000000

a : Analytical non-relativistic energy can be calculated as $E = m_p m_e/(m_p + m_e) \ - 1/(2n^2)$ where $n = v + L$ is the principal quantum number.
b : Analytical relativistic BO energy can be calculated as $E = -1/(2n^2) + 1/(2c^2 n^3)(1 - 0.75/n)$ where $n = v + L$ is the principal quantum number.
c : Analytical non-relativistic BO energy can be calculated as $E = -1/(2n^2)$ where $n = v + L$ is the principal quantum number.

with

$$e(n,k) = \left(1 + \frac{Z^2}{c^2(\sqrt{k^2 - Z^2/c^2} - |k| + n)^2}\right)^{-1/2} \tag{7.58}$$

where n is the principal quantum number in this context. The estimated accuracy of our results is at least six significant digits. The nuclear masses were taken from Ref. [276] and expressed in terms of the electron mass.

Table 7.3: The relativistic fine-structure ground state energies E_{DC} of hydrogen-like ions for nuclear charges $2 \leq Z \leq 10$. E_{nr} presents the energies of the non-relativistic limit. They are compared to the analytical non-relativistic ground state energies.

Isotope	Z	Nuclear Mass[a]	E_{DC} [E_h]	Analyt. E_{DC} [E_h][b]	E_{nr} [E_h]	Analyt. E_{nr} [E_h][c]
$^{3}He^{+1}$	2	5497.8852173	−1.9997426	−1.9997428	−1.99963624	−1.99963628
$^{13}Be^{+3}$	4	23759.609193	−8.001367	−8.001368	−7.99966294	−7.99966329
$^{13}C^{+5}$	6	23697.656980	−18.007871	−18.007875	−17.9992400	−17.9992404
$^{15}N^{+6}$	7	27336.525968	−24.515100	−24.515106	−24.4991035	−24.4991037
$^{15}O^{+7}$	8	27340.915155	−32.02613	−32.02614	−31.9988283	−31.9988296
$^{19}F^{+8}$	9	34622.970060	−40.54258	−40.54259	−40.4988291	−40.4988302
$^{19}Ne^{+9}$	10	34628.308608	−50.06527	−50.06529	−49.9985543	−49.9985561

a : Relative to the electron mass.
b : The analytical reference energies are calculated according to Eqs. (7.57) and (7.58).
c : The analytical reference energies of non-relativistic pre-BO hydrogen-like atoms are calculated as $E_{\mathrm{nr}} = -1/2\ m_n m_e/(m_n + m_e)Z^2$ where m_n is the nuclear mass and Z is the nuclear charge number.

8

Improvements of Computational Efficiency and Scaling Behavior

In this chapter, we present further work focusing on improvements of the computational efficiency and the scaling behavior with respect to the number of particles. Also alternatives for ensuring variational stability in relativistic calculations were investigated.

8.1 Matrix Form of the (Anti-)Symmetrization Operator

The computational cost of imposing the correct particle exchange symmetry scales factorially with respect to the number of particles involved. It is the factor which prohibits the study of systems containing more than a few particles.

The (anti-)symmetrization with respect to some type of particle is presented in Eq. (2.29). It can be recast in product form as

$$\mathcal{A} = (1 - P_{1,2})(1 - P_{1,3} - P_{2,3}) \dots (1 - P_{1,N} - \dots - P_{N-1,N}) \ . \tag{8.1}$$

To show that the two formulations are identical is rather straightforward and can be done by expansion of Eq. (8.1).

Another way of looking at Eq. (8.1) is illustrated in Figure 8.1. Here a particle is added to a system containing N particles with the proper particle

exchange symmetry already imposed. The new $N+1$ wave function is then symmetrized trough the operator

$$\mathcal{A}_N = (1 - P_{1N+1} - \ldots - P_{N\ N+1}) \tag{8.2}$$

which performs all exchanges of the N particles with the $(N+1)^{th}$ particle. Insertion of the resolution of the identity

$$\mathbf{1} = |\mathbf{\Psi}\rangle \langle \mathbf{\Psi}| \tag{8.3}$$

into Eq. (8.1) can be used to form a matrix representation of the (anti-)symmetrization operator as

$$\begin{aligned} \boldsymbol{\mathcal{A}} = \langle \mathbf{\Psi}| (1 - P_{12}) |\mathbf{\Psi}\rangle \langle \mathbf{\Psi}| (1 - P_{13} - P_{23}) |\mathbf{\Psi}\rangle \ldots \\ \times \ldots \langle \mathbf{\Psi}| (1 - P_{1N} - \ldots - P_{N-1\ N}) |\mathbf{\Psi}\rangle \quad . \end{aligned} \tag{8.4}$$

This form scales much more favorable than Eq. (2.29) since it only involves N^2 matrices. The main problem with the matrix form is the resolution of the identity. Treated as an approximation in terms of an auxiliary basis set leads to the obvious problem: How to find the auxiliary basis set? An incomplete auxiliary basis set leads to a false normalization of the state function as

$$\mathbf{1} \geq |\mathbf{\Psi}\rangle \langle \mathbf{\Psi}| \tag{8.5}$$

and therefore

$$\boldsymbol{H} \leq \boldsymbol{\mathcal{A}} \boldsymbol{H}_0 \boldsymbol{\mathcal{A}} \quad \text{and} \quad \boldsymbol{S} \leq \boldsymbol{\mathcal{A}} \boldsymbol{S}_0 \boldsymbol{\mathcal{A}} \tag{8.6}$$

where $\boldsymbol{H}_0$, $\boldsymbol{S}_0$ are the matrix representations of the Hamiltonian and the metric where the particle exchange symmetry is not enforced, and $\boldsymbol{\mathcal{A}}$ is the matrix representation of (anti-)symmetrization operator. Thus, this approximate matrix representation of the (anti-)symmetrization operator leads to variationally unstable calculations because the energy depends on the inverse of $\boldsymbol{S}$.

8.2 Analytical Gradients

Kinghorn has presented a convenient mathematical framework for the derivation of analytical energy gradients [38]. This form of matrix differential calculus was introduced by Magnus and Neudecker [39,40]. It has the advantage of having a well-defined chain rule which is not present in other forms of matrix differential calculus. The derivatives of the energy integral presented

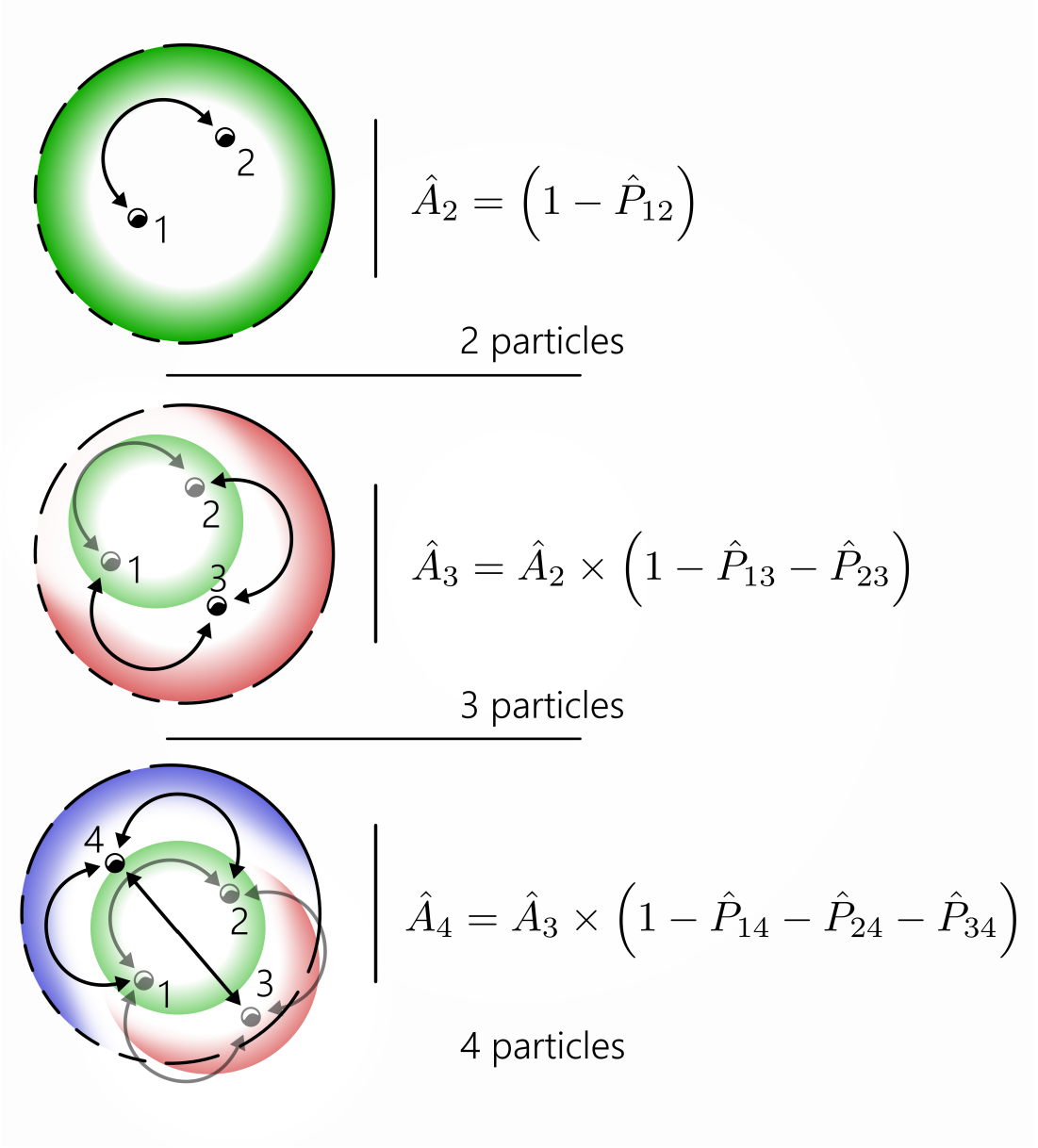

Figure 8.1: Imposing particle exchange symmetry for a system with an increased number of particles. The symmetrization only involves exchanges with the new particle if the original system is already properly symmetrized.

by Kinghorn is not directly applicable to our work, since they use a different parametrization of the correlation matrix $\boldsymbol{A}$ and also do not employ the GVR.

Based on the mathematical framework formulated by Magnus and Neudecker and presented by Kinghorn, we have derived new gradient expressions for the basis functions in Eq. (2.30). These gradients are not included into our computer program BlueBerry, since they are computationally too expensive

to be practical and do not provide us with an advantage over our stochastic sampling routines. Numerical gradients are more efficient than analytical gradients, but do also not provide us with any advantage compared to our stochastic sampling routines.

8.2.1 Matrix Derivative

The general matrix derivative of some function $F(\boldsymbol{A}) : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{p\times q}$ relies on the vec function for a general $n \times m$ matrix

$$\mathrm{vec}\ \boldsymbol{A} = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{n,m-1} & A_{n,m} \end{bmatrix}^{\mathrm{T}} \tag{8.7}$$

and takes the form

$$\boldsymbol{F}'(\boldsymbol{A}) = \frac{\partial \mathrm{vec}\ \boldsymbol{F}(\boldsymbol{A})}{\partial (\mathrm{vec}\ \boldsymbol{A})^{\mathrm{T}}} \ . \tag{8.8}$$

The gradient is

$$\nabla_{\boldsymbol{A}} \boldsymbol{F}(\boldsymbol{A}) = \boldsymbol{F}'(\boldsymbol{A})^{\mathrm{T}} \ . \tag{8.9}$$

Similarly, for square symmetric $n \times n$ matrices, a vech function can be defined as

$$\mathrm{vech}\ \boldsymbol{A} = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{i,i} & A_{i,i+1} & \dots & A_{n,m} \end{bmatrix}^{\mathrm{T}} \ . \tag{8.10}$$

8.2.2 General Method for the Derivation of Gradients

The derivative in Eq. (8.8) is obtained in three steps. The first step is to form $d\boldsymbol{F}(\boldsymbol{X})$ using the product rule

$$d(\boldsymbol{U} \times \boldsymbol{V}) = d\boldsymbol{U} \times \boldsymbol{V} + \boldsymbol{U} \times d\boldsymbol{V} \quad \text{where} \quad \times \in \{\cdot, \otimes\} \ , \tag{8.11}$$

and the following relations for the trace, transpose, inverse and determinant

$$d\mathrm{Tr}\left(\boldsymbol{U}\right) = \mathrm{Tr}\left(d\boldsymbol{U}\right) \ , \tag{8.12}$$

$$d\boldsymbol{U}^{\mathrm{T}} = \left(d\boldsymbol{U}\right)^{\mathrm{T}} \ , \tag{8.13}$$

$$d\det\left(\boldsymbol{U}\right) = \det\left(\boldsymbol{U}\right)\left(\mathrm{vec}(\boldsymbol{U}^{-\mathrm{T}})\right)^{\mathrm{T}} d\mathrm{vec}\ \boldsymbol{U} \ , \tag{8.14}$$

$$d\mathrm{vec}\ \boldsymbol{U}^{-1} = -\left(\boldsymbol{U}^{-\mathrm{T}} \otimes \boldsymbol{U}^{-1}\right) d\mathrm{vec}\ \boldsymbol{U} \ . \tag{8.15}$$

For the derivation of these relations, we refer the reader to either the work by Kinghorn [38] or to the original work by Magnus and Neudecker [39,40].

The next step in the derivation is to form

$$\mathrm{vec}\,(d\boldsymbol{F}(\boldsymbol{A})) = d\mathrm{vec}\,(\boldsymbol{F}(\boldsymbol{A})) = \boldsymbol{F}'(\boldsymbol{A})d\mathrm{vec}(\boldsymbol{A}) \ . \tag{8.16}$$

If $\boldsymbol{F}(\boldsymbol{A})$ is a vector or a scalar, then it is not affected by the vec operation.

The most important relations for this step are

$$\mathrm{vec}\ \boldsymbol{a}\boldsymbol{b}^{\mathrm{T}} = \boldsymbol{b} \otimes \boldsymbol{a} \ , \tag{8.17}$$

$$\mathrm{Tr}\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B}\right) = (\mathrm{vec}\ \boldsymbol{A})^{\mathrm{T}}\,\mathrm{vec}\ \boldsymbol{B} \ , \tag{8.18}$$

$$\left(\boldsymbol{A}^{\mathrm{T}} \otimes \boldsymbol{B}\right)\mathrm{vec}\ \boldsymbol{C} = \mathrm{vec}\,(\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}) \ . \tag{8.19}$$

In the last step we form the derivative of Eq. (8.8) from Eq. (8.16).

8.2.2.1 Structured Matrices

The matrix $\boldsymbol{A}$ is required to be positive-definite. Therefore, a variety of special parametrization schemes for $\boldsymbol{A}$ exist which lead to special structures of this matrix. An important parametrization of $\boldsymbol{A}$ is the Cholesky decomposition:

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} \tag{8.20}$$

in terms of the lower triangular matrix $\boldsymbol{L}$. In order to respect the special structure imposed onto $\boldsymbol{A}$ by Eq. (8.20), the matrix $\mathcal{L}$ is introduced. It is defined by the relation

$$\mathrm{vec}\ \boldsymbol{A} = \mathcal{L}\mathrm{vec}\ L \ . \tag{8.21}$$

Using the chain rule, we can relate any derivative in terms of the original matrix $\boldsymbol{A}$ to $\boldsymbol{L}$ as

$$\boldsymbol{F}'(\boldsymbol{A}(\boldsymbol{L})) = \frac{\partial \boldsymbol{F}(\boldsymbol{A})}{\partial \boldsymbol{A}}\frac{\partial \boldsymbol{A}(\boldsymbol{L})}{\partial \boldsymbol{L}} = \boldsymbol{F}'(\boldsymbol{A})\mathcal{L} \ . \tag{8.22}$$

Similar to Eq. (8.20) the vec and the vech operations can be related in terms of the duplication matrix

$$\mathrm{vech}\ \boldsymbol{A} = \mathcal{D}_n\mathrm{vec}\ \boldsymbol{A} \tag{8.23}$$

and the α parameters in Eq. (2.42) as

$$\mathrm{vec}\ \boldsymbol{\alpha} = \mathcal{Q}\mathrm{vec}\ \boldsymbol{A} \ . \tag{8.24}$$

8.2.3 Matrix Derivative of the Energy

The derivation of the total energy with respect to $\boldsymbol{A}$ is given as [38]

$$\nabla_{\boldsymbol{A}} E = \left(\frac{\partial \mathrm{vech}\ \boldsymbol{H}}{\partial (\mathrm{vec}\ \boldsymbol{A})^{\mathrm{T}}} - E \frac{\partial \mathrm{vech}\ \boldsymbol{S}}{\partial (\mathrm{vec}\ \boldsymbol{A})^{\mathrm{T}}} \right)^{\mathrm{T}} \mathrm{vech}\ \left(2\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}} - \mathrm{diag}\left(\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}} \right) \right) \ , \quad (8.25)$$

if the basis functions are ortho-normal, where $\boldsymbol{H}$ and $\boldsymbol{S}$ are the matrix forms of the Hamiltonian and the overlap operators and $\boldsymbol{c}$ is a vector containing the expansion coefficients of the state of the system. We continue now with the derivatives for the overlap, the kinetic-energy and the potential-energy operators.

8.2.4 Matrix Derivative of the Overlap Integral

The integral expression for the overlap of two ECGs with the GVR is [30]

$$S_{IJ} = D^{3/4} q_I^{-K_I - L/2} q_J^{-K_J - L/2} \sum_{m}^{\min(K_I, K_J)} p_{IJ}^{2m+L} p_{II}^{K_I - m} p_{JJ}^{K_J - m} H_{L, K_I, K_J, m} \quad (8.26)$$

with

$$D = \frac{\det(2\boldsymbol{A}_I) \det(2\boldsymbol{A}_J)}{\det(\boldsymbol{A}_{IJ})^2} \ , \quad (8.27)$$

$$p_{XY} = \boldsymbol{u}_X^T \boldsymbol{A}_{IJ}^{-1} \boldsymbol{u}_X \quad \text{where} \quad X, Y \in \{I, J\} \ , \quad (8.28)$$

$$q_X = \frac{1}{2} \boldsymbol{u}_X^T \boldsymbol{A}_X^{-1} \boldsymbol{u}_X \quad \text{where} \quad X \in \{I, J\} \ . \quad (8.29)$$

The derivative of the first term is obtained through the product rule in Eq. (8.11) and the derivative of the determinant in Eq. (8.14). The resulting expression reads as:

$$\frac{dD}{d\mathrm{vec}(\boldsymbol{A}_I)} = D\left[V(\boldsymbol{A}_I) - 2V(\boldsymbol{A}_{IJ}) \right] \quad \text{with} \quad V(\boldsymbol{A}) = \left(\mathrm{vec}(\boldsymbol{A}^{-\mathbf{T}}) \right)^{\mathrm{T}} \quad (8.30)$$

and the derivative of the quadratic terms read as

$$-\boldsymbol{U}_{XY} = \frac{dp_{XY}}{d\mathrm{vec}(\boldsymbol{A}_X)} = -\left(\boldsymbol{u}_X^T \boldsymbol{A}_{IJ}^{-1} \otimes \boldsymbol{u}_Y^T \boldsymbol{A}_{IJ}^{-1} \right) \quad (8.31)$$

where we have used Eq. (8.15) for the derivative of the inverse and then applied the vec operation. Since p_{XY} is a scalar it is not affected by the vec

operation. Accordingly the derivative of the term of the quasi-normalization is

$$-\frac{1}{2}\boldsymbol{Q}_I = \frac{dq_I}{d\mathrm{vec}(\boldsymbol{A}_I)} = -\frac{1}{2}\left(\boldsymbol{u}_I^T\boldsymbol{A}_I^{-1} \otimes \boldsymbol{u}_I^T\boldsymbol{A}_I^{-1}\right) \ . \tag{8.32}$$

Trivially, the derivative of q_J is

$$\frac{dq_J}{d\mathrm{vec}(\boldsymbol{A}_I)} = 0 \ . \tag{8.33}$$

From relations in Eqs. (8.30) – (8.32) we find the derivative of the overlap integral as

$$\begin{aligned}\frac{dS_{IJ}}{d\mathrm{vec}(\boldsymbol{A}_I)} = S_{IJ}\Bigg(V(\boldsymbol{A}_I) - 2V(\boldsymbol{A}_{IJ}) + \frac{(K_I + L/2)}{2q_I}Q_I - \frac{L}{p_{IJ}}U_{IJ} \\ -\frac{K_I}{p_{II}}U_{II} - \frac{K_J}{p_{JJ}}U_{JJ}\Bigg) + S_{IJ}^{(m)}\left(\frac{1}{p_{II}}U_{II} + \frac{1}{p_{JJ}}U_{JJ} - \frac{2}{p_{IJ}}U_{IJ}\right)\end{aligned} \tag{8.34}$$

with

$$\begin{aligned}S_{IJ}^{(m)} &= D^{3/4}q_I^{-K_I-L/2}q_J^{-K_J-L/2} \\ &\times \sum_m^{\min(K_I,K_J)} m\ p_{IJ}^{2m+L}p_{II}^{K_I-m}p_{JJ}^{K_J-m}H_{L,K_I,K_J,m} \ .\end{aligned} \tag{8.35}$$

8.2.5 Derivative of the Kinetic-Energy Operator

The integral expression for the kinetic energy reads as [30]

$$\begin{aligned}T_{IJ} = D^{3/4}q_I^{-K_I-L/2}q_J^{-K_J-L/2}\sum_m^{\min(K_I,K_J)}[R_{IJ} + (K_I - m)P_{II} \\ +(K_J - m)P_{JJ} + (L + 2m)P_{JJ}]\left(\frac{p_{IJ}p_{IJ}}{p_{II}p_{JJ}}\right)^m H_{K_I,K_J,L,m}\end{aligned} \tag{8.36}$$

with

$$R_{IJ} = \frac{3}{2}\mathrm{Tr}(\boldsymbol{A}_{IJ}^{-1}\boldsymbol{A}_I\boldsymbol{M}\boldsymbol{A}_I) \ , \tag{8.37}$$

$$P_{JJ} = -\boldsymbol{u}_J^T\boldsymbol{A}_{IJ}^{-1}\boldsymbol{A}_I\boldsymbol{M}\boldsymbol{A}_I\boldsymbol{A}_{IJ}^{-1}\boldsymbol{u}_J \ , \tag{8.38}$$

$$P_{II} = -\boldsymbol{u}_I^T\boldsymbol{A}_{IJ}^{-1}\boldsymbol{A}_J\boldsymbol{M}\boldsymbol{A}_J\boldsymbol{A}_{IJ}^{-1}\boldsymbol{u}_I \ , \tag{8.39}$$

$$P_{IJ} = \boldsymbol{u}_I^T\boldsymbol{A}_{IJ}^{-1}\boldsymbol{A}_J\boldsymbol{M}\boldsymbol{A}_I\boldsymbol{A}_{IJ}^{-1}\boldsymbol{u}_J \ . \tag{8.40}$$

The derivative of Eq. (8.37) is obtained through Eq. (8.12) for the trace and Eq. (8.15) for the inverse. It reads as

$$R'_{IJ} = \frac{dR_{IJ}}{d\mathrm{vec}(\boldsymbol{A}_I)} = \frac{3}{2}\mathrm{vec}(\boldsymbol{A}_{IJ}^{-1}\boldsymbol{A}_J\boldsymbol{M}\boldsymbol{A}_J\boldsymbol{A}_{IJ^{-1}}) \ . \tag{8.41}$$

The derivatives of Eqs. (8.38) – (8.40) are

$$\begin{aligned}\boldsymbol{U}_{JJ}P'_{JJ} &= \frac{dP_{JJ}}{d\mathrm{vec}(\boldsymbol{A}_I)} \\ &= \boldsymbol{U}_{JJ}\left[\left(1 \otimes \boldsymbol{A}_I\boldsymbol{M}\boldsymbol{A}_J\boldsymbol{A}_{IJ}^{-1}\right) + \left(\boldsymbol{A}_I\boldsymbol{M}\boldsymbol{A}_J\boldsymbol{A}_{IJ}^{-1} \otimes 1\right)\right] \ , \qquad (8.42) \\ \boldsymbol{U}_{II}P'_{II} &= \frac{dP_{II}}{d\mathrm{vec}(\boldsymbol{A}_I)} \\ &= \left(\boldsymbol{U}_{II}\left(\boldsymbol{1} \otimes \boldsymbol{A}_J\boldsymbol{M}\boldsymbol{A}_J\boldsymbol{A}_{IJ}^{-1}\right) + \boldsymbol{U}_{II}\left(\boldsymbol{A}_J\boldsymbol{M}\boldsymbol{A}_J\boldsymbol{A}_{IJ}^{-1} \otimes 1\right)\right) \qquad (8.43)\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{U}_{IJ}P'_{IJ} &= \frac{dP_{IJ}}{d\mathrm{vec}(\boldsymbol{A}_I)} \\ &= \boldsymbol{U}_{IJ}\left(\left(\boldsymbol{1} \otimes \boldsymbol{A}_J\boldsymbol{M}\boldsymbol{A}_J\boldsymbol{A}_{IJ}^{-1}\right) - \left(\boldsymbol{A}_I\boldsymbol{M}\boldsymbol{A}_J\boldsymbol{A}_{IJ}^{-1} \otimes 1\right)\right) \ . \qquad (8.44)\end{aligned}$$

They are derived using the product rule in Eq. (8.11) and the derivative of the inverse in Eq. (8.15).

In order to facilitate the derivation of the final expression, we continue with the derivatives of the following products

$$\begin{aligned}\frac{d(R_{IJ}p_{IJ}^{L+2m}p_{II}^{K_I-m}p_{JJ}^{K_J-m})}{d\mathrm{vec}(\boldsymbol{A}_I)} &= [\boldsymbol{F}_{R_{IJ}} + \boldsymbol{F}_m \times m] \\ &\quad \times R_{IJ}p_{IJ}^{L+2m}p_{II}^{K_I-m}p_{JJ}^{K_J-m} \ , \qquad (8.45) \\ \frac{d(P_{II}p_{IJ}^{L+2m}p_{II}^{K_I-m-1}p_{JJ}^{K_J-m})}{d\mathrm{vec}(\boldsymbol{A}_I)} &= [\boldsymbol{F}_{P_{II}} + \boldsymbol{F}_m \times m] \\ &\quad \times P_{II}p_{IJ}^{L+2m}p_{II}^{K_I-m-1}p_{JJ}^{K_J-m} \ , \qquad (8.46) \\ \frac{d(P_{JJ}p_{IJ}^{L+2m}p_{II}^{K_I-m-1}p_{JJ}^{K_J-m})}{d\mathrm{vec}(\boldsymbol{A}_I)} &= [\boldsymbol{F}_{P_{JJ}} + \boldsymbol{F}_m \times m] \\ &\quad \times P_{JJ}p_{IJ}^{L+2m}p_{II}^{K_I-m}p_{JJ}^{K_J-m-1} \qquad (8.47)\end{aligned}$$

and

$$\begin{aligned}\frac{d(P_{JJ}p_{IJ}^{L+2m}p_{II}^{K_I-m-1}p_{JJ}^{K_J-m})}{d\mathrm{vec}(\boldsymbol{A}_I)} &= [\boldsymbol{F}_{P_{IJc}} + \boldsymbol{F}_{P_{IJm}} \times m] \\ &\quad \times P_{IJ}p_{IJ}^{L+2m-1}p_{II}^{K_I-m}p_{JJ}^{K_J-m} \ , \qquad (8.48)\end{aligned}$$

where

$$\boldsymbol{F}_{R_{IJ}} = \frac{\boldsymbol{R}'_{IJ}}{R_{IJ}} - \frac{L\boldsymbol{U}_{IJ}}{p_{IJ}} - \frac{K_I\boldsymbol{U}_{II}}{p_{II}} - \frac{K_J\boldsymbol{U}_{JJ}}{p_{JJ}} \ , \tag{8.49}$$

$$\boldsymbol{F}_{P_{II}} = \frac{\boldsymbol{U}_{II}P'_{II}}{P_{II}} - \frac{L\boldsymbol{U}_{IJ}}{p_{IJ}} - \frac{(K_I-1)\boldsymbol{U}_{II}}{p_{II}} - \frac{K_J\boldsymbol{U}_{JJ}}{p_{JJ}} \ , \tag{8.50}$$

$$\boldsymbol{F}_{P_{JJ}} = -\frac{\boldsymbol{U}_{JJ}P'_{JJ}}{P_{JJ}} - \frac{L\boldsymbol{U}_{IJ}}{p_{IJ}} - \frac{K_I\boldsymbol{U}_{II}}{p_{II}} - \frac{(K_J-1)\boldsymbol{U}_{JJ}}{p_{JJ}} \ , \tag{8.51}$$

$$\boldsymbol{F}_{P_{IJ}} = \frac{\boldsymbol{U}_{IJ}P'_{IJ}}{P_{IJ}} - \frac{(L-1)\boldsymbol{U}_{IJ}}{p_{IJ}} - \frac{K_I\boldsymbol{U}_{II}}{p_{II}} - \frac{K_J\boldsymbol{U}_{JJ}}{p_{JJ}} \ , \tag{8.52}$$

$$\boldsymbol{F}_m = \frac{\boldsymbol{U}_{II}}{p_{II}} + \frac{\boldsymbol{U}_{JJ}}{p_{JJ}} - \frac{2\boldsymbol{U}_{IJ}}{p_{IJ}} \ . \tag{8.53}$$

We can then use Eqs. (8.45) – (8.48) to obtain the following derivative

$$\begin{aligned}&\frac{d}{d\mathrm{vec}A_I}\left[R_{IJ} + (K_I-m)\frac{P_{II}}{p_{II}} + (K_J-m)\frac{P_{JJ}}{p_{JJ}}\right.\\ &\left.+(L+2m)\frac{P_{IJ}}{p_{IJ}}\right]\left(\frac{p_{IJ}p_{IJ}}{p_{II}p_{JJ}}\right)^m = \left[\boldsymbol{F}_0 + \boldsymbol{F}_1 m + \boldsymbol{F}_2 m^2\right]\left(\frac{p_{IJ}p_{IJ}}{p_{II}p_{JJ}}\right)^m\end{aligned} \tag{8.54}$$

with

$$\boldsymbol{F}_0 = R_{IJ}\boldsymbol{F}_{R_{IJ}} + K_I\boldsymbol{F}_{P_{II}}\frac{P_{II}}{p_{II}} + K_J\boldsymbol{F}_{P_{JJ}}\frac{P_{JJ}}{p_{JJ}} + L\boldsymbol{F}_{P_{IJ}}\frac{P_{IJ}}{p_{IJ}} \ , \tag{8.55}$$

$$\begin{aligned}\boldsymbol{F}_1 = {} & R_{IJ}\boldsymbol{F}_m + K_I\boldsymbol{F}_m\frac{P_{II}}{p_{II}} - \boldsymbol{F}_{P_{II}}\frac{P_{II}}{p_{II}} + K_J\boldsymbol{F}_m\frac{P_{JJ}}{p_{JJ}}\\ & - \boldsymbol{F}_{P_{JJ}}\frac{P_{JJ}}{p_{JJ}} + L\boldsymbol{F}_m\frac{P_{IJ}}{p_{IJ}} + 2\boldsymbol{F}_{P_{IJ}}\frac{P_{IJ}}{p_{IJ}} \ ,\end{aligned} \tag{8.56}$$

$$\boldsymbol{F}_2 = \left[2\frac{P_{IJ}}{p_{IJ}} - \frac{P_{II}}{p_{II}} - \frac{P_{JJ}}{p_{JJ}}\right]\boldsymbol{F}_m \ . \tag{8.57}$$

We are now ready to derive the final expression for the derivative of the kinetic-energy integral. By combining Eqs. (8.30) – (8.32) from the derivative of overlap integral together with Eq. (8.54) leads to the final expression for the derivative of the kinetic-energy integral:

$$\begin{aligned}&\frac{d}{d\mathrm{vec}(A_I)}T_{IJ} = T_{IJ}[V(\boldsymbol{A}_I) - 2V(\boldsymbol{A}_{IJ})] + T_{IJ}\frac{K_I+L/2}{2q_I}\boldsymbol{Q}_I + D^{3/4}q_I^{-K_I-L/2}\\ &\times q_J^{-K_J-L/2}\sum_m^{\min(K_I,K_J)}\left[\boldsymbol{F}_0 + \boldsymbol{F}_1 m + \boldsymbol{F}_2 m^2\right]\left(\frac{p_{IJ}p_{IJ}}{p_{II}p_{JJ}}\right)^m H_{K_I,K_J,L,m} \ .\end{aligned} \tag{8.58}$$

8.2.6 Derivative of the Coulomb Interaction Integral

Since the Coulomb potential only depends on the distance between particles i and j, it is advantageous to introduce the transformed coordinates [30]

$$\boldsymbol{U} = \mathbf{1} - \boldsymbol{I}_{ij} \quad \text{where} \quad \boldsymbol{I}_{ij} = 1 \quad \text{if} \quad j > i \quad \text{and else} \quad 0 \tag{8.59}$$

which substitutes the coordinate of the i^{th} particle which the distance of particles i and j. After the transformation, the integral expression of the Coulomb integral reads as [30]

$$\begin{aligned} V_i = \sqrt{\frac{2}{\pi}} G^{3/4} q_I^{-K_I - L/2} q_J^{-K_J - L/2} \sum_{m}^{L+K_I+K_J} \sum_{i}^{m} \sum_{j}^{m-i} \sum_{n} c_{IJ}^{-1-m} t_I^{m+i-j} t_J^{m-i+j} \\ \times \tau_{II}^{-i-n+K_I} \tau_{JJ}^{-j-n+K_J} \tau_{IJ}^{-m+i+j+2n+L} H_{m,i,j,n}^{L,K_I,K_J} \end{aligned} \tag{8.60}$$

with

$$G = \frac{\det(2\boldsymbol{A}_I)\det(2\boldsymbol{A}_J)}{\det\boldsymbol{\Gamma}\det\boldsymbol{\Gamma}} \ , \tag{8.61}$$

$$t_X = (\boldsymbol{u}_X)_1 - \boldsymbol{\gamma}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{\omega}_X \quad \text{with} \quad X \in \{I, J\} \ , \tag{8.62}$$

$$\tau_{XY} = \boldsymbol{\omega}_X^T\boldsymbol{\Gamma}^{-1}\boldsymbol{\omega}_Y \quad \text{with} \quad X, Y \in \{I, J\} \ , \tag{8.63}$$

$$c_{IJ} = A_{12} - \boldsymbol{\gamma}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma} \ . \tag{8.64}$$

$\boldsymbol{\Gamma}$, $\boldsymbol{\gamma}$, and A_{12} are defined as in terms of $\boldsymbol{A}_{IJ}$. $\boldsymbol{\Gamma}$ is obtained by eliminating the i^{th} row and column from $\boldsymbol{A}_{IJ}$. $\boldsymbol{\gamma}$ is the j^{th} column of $\boldsymbol{A}_{IJ}$ with the i^{th} element removed. Last, A_{12} is the i^{th} element of the j^{th} column in $\boldsymbol{A}_{IJ}$.

$\boldsymbol{\omega}_X$ and $(\boldsymbol{u}_X)_1$ are defined in terms of $\boldsymbol{u}_X$ where $\boldsymbol{\omega}_X$ is $\boldsymbol{u}_X$ with the i^{th} element eliminated and $(\boldsymbol{u}_X)_1$ is the i^{th} element of $\boldsymbol{u}_X$.

Before we can continue with the derivation of the derivative of the Coulomb interaction integral, we need to formulate matrix expressions for $\boldsymbol{\Gamma}$, $\boldsymbol{\gamma}$, A_{12}, $\boldsymbol{\omega}_X$ and $(\boldsymbol{u}_X)_1$. For this, we introduce the $N \times (N-1)$ matrix $\boldsymbol{R}_i$ which is defined as the $N \times N$ identity matrix with the i^{th} column eliminated. We then find

$$\boldsymbol{\Gamma} = \boldsymbol{R}_i^T\boldsymbol{A}_{IJ}\boldsymbol{R}_i \ . \tag{8.65}$$

Furthermore, we introduce the N vector $\boldsymbol{s}_i$ with the element at position q defined as $(\boldsymbol{s}_i)_q = \delta_{iq}$. The remaining terms are then defined as

$$\boldsymbol{\gamma} = \boldsymbol{R}_i^T\boldsymbol{A}_{IJ}\boldsymbol{s}_i \ , \tag{8.66}$$

$$A_{12} = \boldsymbol{s}_i^T\boldsymbol{A}_{IJ}\boldsymbol{s}_i \ , \tag{8.67}$$

$$\boldsymbol{\omega} = \boldsymbol{R}_i^T\boldsymbol{u} \ , \tag{8.68}$$

$$(u)_1 = \boldsymbol{s}_i\boldsymbol{u} \ . \tag{8.69}$$

Eq. (8.60) contains the inverse of $\mathbf{\Gamma}$ and since $\boldsymbol{R}_i$ has no inverse, it is not possible to rearrange Eq. (8.65) in terms of the inverse of the individual matrices. Therefore, we have to use the chain rule to obtain the derivative of the inverse of $\mathbf{\Gamma}$:

$$d\mathbf{\Gamma}^{-1} = -\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^{\mathrm{T}} d\boldsymbol{A}_I \boldsymbol{R}_i \mathbf{\Gamma}^{-1} \ . \tag{8.70}$$

The derivative of c_{IJ} then reads

$$\begin{aligned} c'_{IJ} = \frac{dc_{IJ}}{d\mathrm{vec}(\boldsymbol{A}_I)} = (\boldsymbol{s}_i^T \otimes \boldsymbol{s}_i) + (\boldsymbol{\gamma}^T\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^T \otimes \boldsymbol{s}_i^T) \\ -(\boldsymbol{\gamma}^T\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^T \otimes \boldsymbol{\gamma}^T\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^T) + (\boldsymbol{s}_i^T \otimes \boldsymbol{\gamma}^T\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^T) \ . \end{aligned} \tag{8.71}$$

The derivatives of Eqs. (8.62) and (8.63) are

$$\begin{aligned} t'_X &= \frac{dt_X}{d\mathrm{vec}(\boldsymbol{A}_I)} = \\ &= (\boldsymbol{\omega}_X^T\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^T \otimes \boldsymbol{\gamma}^T\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^T) - (\boldsymbol{\omega}_X^T\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^T \otimes \boldsymbol{s}_i^T) \quad \text{with} \quad X \in \{I, J\} \end{aligned} \tag{8.72}$$

and

$$\tau'_{XY} = \frac{\tau_{XY}}{d\mathrm{vec}(\boldsymbol{A}_I)} = (\boldsymbol{\omega}_X^T\mathbf{\Gamma}\boldsymbol{R}_i^T \otimes \boldsymbol{\omega}_Y^T\mathbf{\Gamma}\boldsymbol{R}_i^T) \quad \text{with} \quad X, Y \in \{I, J\} \ . \tag{8.73}$$

The final expression for the derivative of the Coulomb interaction is

$$\begin{aligned} \frac{dV_i}{d\mathrm{vec}(\boldsymbol{A}_I)} = V_i\left[V(\boldsymbol{A}_I) - 2V(\boldsymbol{R}_i\mathbf{\Gamma}^{-1}\boldsymbol{R}_i^T)\right] - V_i\frac{K_I + L/2}{2q_I}\boldsymbol{Q}_I \\ +\sqrt{\frac{2}{\pi}}G^{3/4}q_I^{-K_I-L/2}q_J^{-K_J-L/2}\sum_{m}^{L+K_I+K_J}\sum_{i}^{m}\sum_{j}^{m-i}\sum_{n}\Bigg[-\frac{1+m}{c_{IJ}}\boldsymbol{c}'_{IJ} \\ +\frac{m+i-j}{t_I}\boldsymbol{t}'_I + \frac{m-i+j}{t_J}\boldsymbol{t}'_J + \frac{K_I-i-n}{\tau_{II}}\boldsymbol{\tau}'_{II} \\ +\frac{K_J-j-n}{\tau_{JJ}}\boldsymbol{\tau}'_{JJ} + \frac{L-m+i+j+2n}{\tau_{IJ}}\boldsymbol{\tau}'_{IJ}\Bigg] \times \\ c_{IJ}^{-1-m}t_I^{m+i-j}t_J^{m-i+j}\tau_{II}^{-i-n+K_I}\tau_{JJ}^{-j-n+K_J}\tau_{IJ}^{-m+i+j+2n+L}H_{m,i,j,n}^{L,K_I,K_J} \ . \end{aligned} \tag{8.74}$$

9

Conclusion and Outlook

9.1 Conclusion

Explicitly correlated methods have proven to be highly successful in theoretical physics and chemistry for the study of many-body systems. The accuracy of variational non-relativistic pre-BO calculations employing ECGs are unparalleled regarding both the energy and molecular properties. The relativistic descriptions of such systems using similar variational approaches has long been under-developed and relativity was primarily included through perturbation-based approaches. In this work, we discussed the main obstacles arising when one aims at developing a variational first-quantized relativistic many-1/2-fermion framework: ensuring both variational stability and translational invariance. Furthermore, we have presented solutions to these problems and illustrated their effectiveness in terms of numerical examples.

In chapter 6, we focused on variational collapse in variational Dirac–Coulomb calculations. Variational collapse occurs due to the unboundedness of the Dirac Hamiltonian and the finite size of the basis set [129]. In this work, it is avoided by an explicitly correlated kinetic-balance condition. In order to evaluate the integrals, we had to recast the explicitly correlated basis functions from ECGs with the GVR to CECGs. This is necessary because the ECGs do not maintain their general form after enforcing the kinetic-balance condition. The integral expressions for CECGs are derived according to the work presented by Saito and Suzuki [267] and the work presented by Shiozaki [270].

The problem of translational invariance is not solved by the introduction of a TICC transformed Hamiltonian as it is generally done for the non-relativistic

Schrödinger Hamiltonian. The relativistic equivalent to the center-of-mass frame is the center-of-momentum frame and an accordingly transformed Hamiltonian can be formulated. Also, the transformation of the kinetic balance condition is, in principle, not an issue.

In chapter 4, we showed how translational effects can be eliminated from integral expressions using a special parametrization. This leads to our framework of translationally invariant integrals which allows us to perform calculations in LFCCs. This framework can be applied equally to relativistic and non-relativistic calculations of energies and also to calculations of molecular properties. We have therefore developed a common description of translational invariance for both the relativistic and non-relativistic framework. This facilitated the implementation of our code immensely and allowed us to use non-relativistic parameter sets in the relativistic calculations as they are related to the non-relativistic limit through our explicitly correlated kinetic-balance condition.

We have applied our findings to calculate transition dipole moments for the hydrogen molecule using our non-relativistic framework, presented in chapter 5. Furthermore, we illustrated in chapter 7 that we can formulate a translationally invariant and variationally stable many-1/2-fermion theory by reproducing the fine-structure spectrum of atomic hydrogen and accurate ground-state energies for hydrogen-like ions. Since the Lamb shift is not an inherent part of the Dirac–Coulomb Hamiltonian, it was not part of our fine-structure calculations.

9.2 Outlook

Future work should primarily focus on the extension of our framework for systems of more than two particles. This requires the derivation of an appropriate kinetic balance for each system size. While the basic framework is completed, there is still work left regarding extensions of both the relativistic and non-relativistic framework. In this section we present ideas for further projects.

9.2.1 Visualization of pre-Born–Oppenheimer Particle Densities

The visualization of electron densities is an important tool in quantum chemistry. They help us to understand chemical behavior and physical properties. For pre-BO wave functions, there currently exists no method which visualizes many-particle densities. The main issues are the rotational properties of the wave function. While the molecular structure exists in a pre-BO wave

function, its features need to be individually extracted [82] unlike in BO frameworks, where the structure is an explicit part of the theory. Combining these individually extracted features into one particle density should be studied.

Only the translationally invariant degrees of freedom are of interest for chemists. Translational and rotational degrees of freedom therefore need to be eliminated from the particle densities. The Z-Matrix [277] is a means of storing BO structures independent of their position in space and angle of rotation. It encodes all relevant information in terms of bond length, bond angles, and dihedral angles.

The concept of the Z-Matrix can be used to develop a similar formalism for pre-BO particle densities. The density of a two-particle system is then one-dimensional and defined by the distance between the two particles. If a third particle is added to the system, then the next particle density is two-dimensional, defined by the distance to the second particle, and the angle with the first and second particle. For any further particle which is added to the system, the particle density is three-dimensional. Figure 9.1 illustrates the process how a particle density for pre-BO systems is constructed based on the concept of the Z-Matrix for two and three particles.

The one-particle density of the first particle, $\rho_1(\boldsymbol{r}_1)$, is simply a Dirac delta function,

$$\rho_1(r) = \delta(r) \ , \tag{9.1}$$

located at the origin. The two-particle density, $\rho_2(R)$, of the first and second particle only depends in the inter-particle distance,

$$\rho_2(r) = \langle \boldsymbol{\Psi} | \, \delta(r_{12} - r) \, | \boldsymbol{\Psi} \rangle \ . \tag{9.2}$$

The three-particle density, $\rho_3(R)$, of the first three particles is then,

$$\rho_3(r, \phi) = \langle \boldsymbol{\Psi} | \, \delta(r_{13} - r) \delta(\phi_{312} - \phi) \, | \boldsymbol{\Psi} \rangle \ , \tag{9.3}$$

where r_{13} is the distance between the first and third particle, and ϕ_{312} is the angle between the third and second particle around the first particle.

The resulting particle densities can be interpreted as the densities as-seen from the first particle using the second an third particles as a spatial reference. Thus, the density distribution of the first particle is a Dirac-delta function. The density distribution of the second particle is a linear distribution since it uses the first particle as a reference and the particle-particle distance is the

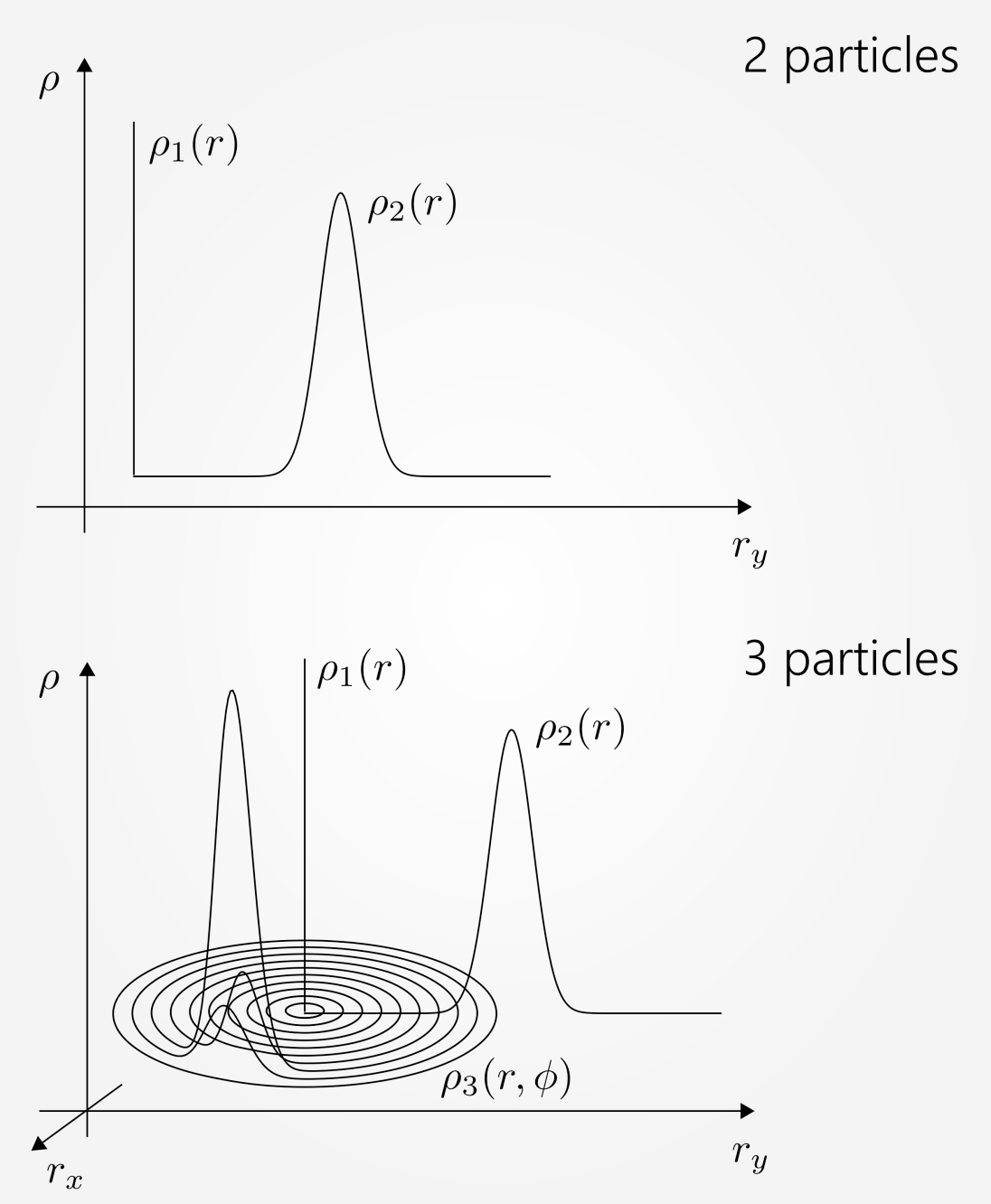

Figure 9.1: Generation of pre-BO particle densities for two and three particle. The top part illustrates a particle density distribution for two particles. The middle part illustrates the particle density distributions for a three-particle system.

only relevant information remaining. The density distribution of the third particle then uses the second particle as the angular reference which leads to a two-dimensional density distribution.

The main challenge is the derivation of computationally feasible integral expressions for the angular and dihedral particle densities. For ECGs with the GVR, the angular integral expressions are computationally expensive. CECGs might provide a basis for more computationally feasible integral expressions.

9.2.2 Hybrid Cartesian-Coordinates Sampling Method

A TICC-parametrized matrix $\boldsymbol{A}$ can be transformed to LFCC and vice verse using Eqs. (2.37) and (2.38). This can be exploited to form a framework where any TICC can be used for the stochastic sampling. Different TICC might accentuate different correlation paths and increase sampling efficiency. For the calculation of the integrals, the convenient LFCC framework can be used to eliminate the translational effects.

9.2.3 Matrix Form of the (Anti-)Symmetrization Operator

In section 8.1, we have presented an alternative scheme for enforcing the particle-exchange symmetry, based on a matrix representation of the (anti-)symmetrization operator. The operator exhibits polynomial scaling with respect to the number of particles. However, as it involves the resolution of the identity approximation, variational collapse occurs. Another approach for the formulation of a matrix form of the (anti-)symmetrization operator is based on the relation:

$$\boldsymbol{S} = \boldsymbol{\mathcal{A}}^T \boldsymbol{S}_0 \boldsymbol{\mathcal{A}} \ , \tag{9.4}$$

where $\boldsymbol{S}$ is the matrix representation of the metric with the correct particle exchange symmetry imposed, $\boldsymbol{S}_0$ is the matrix of the metric with no particle exchange symmetry imposed and $\boldsymbol{\mathcal{A}}$ is the matrix representation of the (anti-)symmetrization operator. An expression for $\boldsymbol{\mathcal{A}}$ can be obtained by starting from

$$\boldsymbol{S} = \boldsymbol{\mathcal{A}}^T \boldsymbol{S}_0 \boldsymbol{\mathcal{A}} = (\boldsymbol{S}_0^{1/2} \boldsymbol{\mathcal{A}})^T \boldsymbol{S}_0^{1/2} \boldsymbol{\mathcal{A}} \ . \tag{9.5}$$

Taking the square root of both sides leads to

$$\boldsymbol{S}^{1/2} = \boldsymbol{S}_0^{1/2} \boldsymbol{\mathcal{A}} \ . \tag{9.6}$$

Acting with the inverse of $\boldsymbol{S}_0^{1/2}$ from the left on both sides of the equation leads to the final expressions

$$\boldsymbol{\mathcal{A}} = \boldsymbol{S}_0^{-1/2} \boldsymbol{S}^{1/2} \ . \tag{9.7}$$

This method exhibits factorial scaling with respect to the number of particles since $\boldsymbol{S}$ is calculated using the exact (anti-)symmetrization operator. However, only (computationally cheap) overlap integrals have to be evaluated to determine $\boldsymbol{\mathcal{A}}$ and the number of integrals of the Hamiltonian becomes independent of the number of particles. Furthermore, this will simplify the problem of finding an auxiliary basis set for the matrix form of the (anti-)symmetrization operator presented in section 8.1 if Eq. (8.4) is applied to Eq. (9.5).

An important question regarding the matrix representation of the (anti-)symmetrization operator is whether different operators have different $\boldsymbol{\mathcal{A}}$.

9.2.4 Hyperfine-Structure of Atoms and Molecules

So far, we have only studied the fine-structure spectrum of 1/2-fermion-binaries in terms of the Dirac–Coulomb Hamiltonian. Further steps, should focus on the extension to many-1/2-fermion systems. This requires the implementation of the kinetic-balance condition for more than two 1/2-fermions either complete or in an approximate way.

Furthermore, future developments should focus on the inclusion of magnetic and retardation terms. This can be achieved in terms of the Breit interaction as presented in Chapter 7.

A

List of Abbreviations

BO Born–Oppenheimer approximation

CC Coupled cluster

CCR Complex-coordinate rotation

CECG Explicitly Correlated Gaussian with Cartesian polynomial prefactors

CI Configuration interaction

CMCC Center-of-mass Cartesian coordinate

DHF Dirac–Hartree–Fock

DKH Douglas–Kroll–Hess

ECG Explicitly correlated Gaussian function

GTO Gaussian-type orbital

GVR Global vector representation

IPM Independent particle model

LFCC Laboratory-fixed Cartesian coordinates

MCSCF Multi-configuration self-consisting field

MP Møller–Plesset perturbation theory

pre-BO pre-Born–Oppenheimer

Appendix A List of Abbreviations

QED Quantum electrodynamics

SCF Self-consisting field

STO Slater-type orbital

TICC Translationally invariant Cartesian coordinates

TICMCC Translationally invariant and center-of-mass Cartesian coordinates

TII Translationally invariant integrals

B

Mathematical Relations

B.1 Tracy-Singh Product

In this part, we introduce the Tracy-Singh product [278] and present its relation to the Kronecker product. The Tracy-Singh product is defined as

$$\boldsymbol{A}_{\text{tsp}} = \boldsymbol{B} \boxtimes \boldsymbol{C} = [(\boldsymbol{B}_{ij} \otimes \boldsymbol{C}_{uv})] = \begin{bmatrix} (\boldsymbol{B}_{11} \otimes \boldsymbol{C}_{uv}) & \cdots & (\boldsymbol{B}_{1n} \otimes \boldsymbol{C}_{uv}) \\ \vdots & & \\ (\boldsymbol{B}_{m1} \otimes \boldsymbol{C}_{uv}) & \cdots & (\boldsymbol{B}_{mn} \otimes \boldsymbol{C}_{uv}) \end{bmatrix} \tag{B.1}$$

where $\boldsymbol{B} = [\boldsymbol{B}_{ij}]$ and $\boldsymbol{C} = [\boldsymbol{C}_{uv}]$ are two matrices of dimension $m \times n$ and $p \times q$, respectively. They are partitioned block-wise in terms of the matrices $\boldsymbol{B}_{ij}$ and $\boldsymbol{C}_{uv}$. $\boldsymbol{A}_{\text{tsp}}$ is a matrix of dimension $mp \times nq$. It is partitioned block-wise with the elements being the matrices $(\boldsymbol{B}_{ij} \otimes \boldsymbol{C}_{uv})$. The Tracy-Singh product may be considered a more general form of the Kronecker product

$$\boldsymbol{A}_{\text{kp}} = \boldsymbol{B} \otimes \boldsymbol{C} = [(b_{ij} c_{uv})] = \begin{bmatrix} b_{11}\boldsymbol{C} & \cdots & b_{1n}\boldsymbol{C} \\ \vdots & & \\ b_{m1}\boldsymbol{C} & \cdots & b_{mn}\boldsymbol{C} \end{bmatrix} \tag{B.2}$$

where b_{ij} and c_{uv} are the matrix elements of $\boldsymbol{B}$ and $\boldsymbol{C}$, respectively. $\boldsymbol{A}_{\text{kp}}$ is a matrix of dimension $mp \times nq$. The two matrices $\boldsymbol{A}_{\text{tsp}}$ and $\boldsymbol{A}_{\text{kp}}$ are identical in the case that $\boldsymbol{B}$ and $\boldsymbol{C}$ are not partitioned (or partitioned into 1×1 blocks). Generally the two products are related through a permutation of the row and column space of either matrix [279–281]

$$\boldsymbol{P}^T (\boldsymbol{B}_1 \otimes \ldots \otimes \boldsymbol{B}_n) \boldsymbol{Q} = \boldsymbol{B} \boxtimes \ldots \boxtimes \boldsymbol{B}_n \tag{B.3}$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are the permutation matrices for the row and the column space and n is the number of matrices involved. For vectors we find the relation

$$\boldsymbol{P}^{\mathrm{T}}(\boldsymbol{v}_1 \otimes \ldots \otimes \boldsymbol{v}_n) = \boldsymbol{v}_1 \boxtimes \ldots \boxtimes \boldsymbol{v}_n \tag{B.4}$$

where $\boldsymbol{v}$ and $\boldsymbol{t}$ are two vectors of arbitrary dimension. The partitioning of the matrices and vectors depends on the permutation matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$. If all matrices are square and symmetrically partitioned, we find that the two permutation matrices are identical [279]

$$\boldsymbol{P} = \boldsymbol{Q} \tag{B.5}$$

and the two products are related through a unitary transformation.

B.2 Row Reduction and Row Reduced Echelon Form

Systems of linear equations are conveniently solved by first representing them in matrix form

$$\boldsymbol{A} \cdot \boldsymbol{x} - \boldsymbol{b} = 0 \ , \tag{B.6}$$

where $\boldsymbol{A}$ is a matrix containing the linear factors. $\boldsymbol{x}$ is a vector and contains the values which are to be determined and $\boldsymbol{b}$ is a vector containing the constant factors of the linear system.

A reliable method of solving such a linear system is row reduction, i.e., Gaussian elimination. It involves performing a series of operations on the augmented form

$$\boldsymbol{A}_{\mathrm{aug}} = \left[\boldsymbol{A}|\boldsymbol{b}\right] \tag{B.7}$$

until it is in row-reduced echelon form. The row-reduced echelon form is

$$\boldsymbol{A}_{\mathrm{rre}} = \left[\mathbf{1}|\boldsymbol{b}'\right] \tag{B.8}$$

for systems with a unique solution. Possible operations are permutation of two rows, multiplication of individual rows with a constant scalar factor and evaluating the difference of two rows.

C

BlueBerry Reference Manual

In this chapter, we present the principal ideas behind the implementation of our pre-BO framework called BlueBerry. It is implemented in C++ and heavily relies on third-party libraries in order to provide reliability and to minimize maintenance of the code.

C.1 YAML File Format

All input and output is standardized using the YAML file format [282]. It was developed for the purpose of data serialization and third-party libraries exist for a variety of programming languages. It is based on the idea, that all information can either be stored as lists or associative arrays or as combinations of the two. Listing C.1 shows a simple example for a list. It contains three elements, each starts with a "-". Also, note that each YAML file starts with a --- as the first line.

Listing C.1: A simple example for a YAML formatted list containing three elements.

```
---
- first element
- second element
- third element
```

Listing C.2 shows an example for an associative array. Here, each entry consists of a key and a value, separated by a ": ". The space is important and cannot be omitted. One of the peculiarities of YAML is that the order of elements in an associative array is not fixed and any permutation of the entries is considered identical.

Listing C.2: A simple example for a YAML formatted associative array containing three elements. The key and value are separated by ": " .

```
---
first key: first value
second key: second value
third key: third value
```

Each element of an array or a list can again be an array or a list. Sublists and Subarrays are organized in terms of their indentation. Listing C.3 presents an example for a simple BlueBerry input file.

Listing C.3: A simple example for a YAML formatted BlueBerry input file. The sublists and subarrays are defined through their indentation.

```
---
composition:
    particles:
      - type: E
        number: 1
        s: 1
        ms: 1
      - type: H1
        number: 1
        s: 1
        ms: 1
    angular momentum:
      j: 0
      mj: 0
    cA: 1
    born oppenheimer: false
size: 100
relativistic: true
polynomials: true
loewdin: 1.0e-12
potential:
    - BBCCoulomb
kinetic: BBCKinetic
overlap: BBCOverlap
```

Here, we see how the indentation leads to an easily readable structure.

C.1.1 XML vs. YAML

The XML file format is widely employed for data serialization. In quantum chemistry, its CML variant [283–289] serves as a basis of large scale automation of calculations and big-data evaluation. But since XML was developed for a different purpose, namely as a markup language, it has certain drawbacks. The main issue associated with XML is its verbosity. Earlier versions of BlueBerry employed XML. But its hard-to-read style made the generation of input files tedious and its verbosity led to large files for storing basis set information. We will therefore not present any technical information about XML. Information about the standard can be found on the website of the W3C [290].

The same information presented in Listing C.3 can also be formatted according to the XML standard. Yet, this form is much more verbose as it can be seen from Listing C.4.

Listing C.4: An XML formatted BlueBerry input file. The tag structure leads to a lot of noise

```
<?xml version="1.0"?>
<input>
  <composition>
    <particles>
      <particle>
        <type>E</type>
        <number>1</number>
        <s>1</s>
        <ms>1</ms>
      <particle>
      <particle>
        <type>H1</type>
        <number>1</number>
        <s>1</s>
        <ms>1</ms>
      <particle>
    </particles>
    <angular_momentum>
      <j>0</j>
      <mj>0</mj>
    </angular_momentum>
    <cA>1</cA>
    <born-oppenheimer>false</born-oppenheimer>
```

```
  </composition>
  <size>100</size>
  <relativistic>true</relativistic>
  <polynomials>true</polynomials>
  <loewdin>1.0e-12</loewdin>
  <potentials>
    <potential>
      BBCCoulomb
    </potential>
  </potentials>
  <kinetic>BBCKinetic</kinetic>
  <overlap>BBCOverlap</overlap>
</input>
```

In XML, all information is organized through a tag structure. This tag structure leads to a large amount of noise. This noise is responsible for the large file size and its hard-to-read structure.

C.2 External Resources

BlueBerry depends on externally stored data. This includes particle specific parameters such as the mass, charge and spin of the different isotopes and particles. Furthermore, precalculated values are stored to facilitate the integral evaluation and polynomial generation. All files are stored in the resources directory and in the YAML file format. The files containing resource information are

particleData.yml: Stores particle specific parameters such as mass, charge and spin.

F2Values.yml: Precalculated values of F_{KL} in Eq. (4.3). Order of keys is L, K.

HValues.yml: Precalculated values of $H_{LK_IK_Jm}$ in Eq. (4.2). Order of keys is L, K_J, K_I, m.

H1Values.yml: Precalculated values of $H_1(m, K_1, K_2, L)$ in Eq. (5.25). Order of keys is L, K_J, K_I, m.

H2Values.yml: Precalculated values of $H_2(m, K_1, K_2, L)$ in Eq. (5.26). Order of keys is L, K_J, K_I, m.

GValues2.yml: Precalculated values of $K(m, \lambda)$ in Eq. (7.52). Order of keys is m, λ.

Factorials.yml: Precalculated Factorials.

Binomials.yml: Precalculated Binomials.

C.3 Dependencies

BlueBerry relies on a number of third-part libraries in order to increase stability and performance while reducing the effort of maintenance. The libraries are:

Armadillo: C++ linear algebra library, capable of interfacing to BLAS implementations [266]. It provides a number of convenient classes for handling vectors, matrices and cubes and contains many routines for most important linear algebra functions such as inverse, eigensolvers etc. It will be highly efficient if the appropriate BLAS implementations are linked.

BLAS: Basic Linear Algebra Subprograms for which a variety of implementations exist such as OpenBlas [291, 292] (based on GotoBLAS2 1.13 BSD version — freely available under Ref. [293]) or the MKL library distributed by intel [294].

yaml-cpp: A library capable of parsing and emitting yaml files. Freely available under Ref. [295].

gsl: GNU Scientific Library [296]. Used for the efficient calculation of Clebsh-Gordan coefficients.

libb64: A library providing the functionality of base64 two-way text-to-binary conversion [297]. This makes it possible for BlueBerry to store parameter set information in binary. Freely available under Ref. [298].

C.4 BBAlphaOperator Class Reference

BBAlphaOperator is an implementation of a component of the $\boldsymbol{\alpha}_i(j)$ three-vector (cf. Eq. (3.2)) as part of the many-fermion Dirac Hamiltonian where j is the particle index. It acts on the **BBRBasisFunction** class to generate a spin-transformed instance of the **BBRBasisFunction**. The matrix is formed using the **BBSigmaOperator** class. It is used in the integral expressions for

the Breit and the Gaunt operators. It also contains operators for the coupling of several **BBAlphaOperator** instances.

`#include <BBAlphaOperator.h>`

Public Member Functions

- **BBAlphaOperator**(**BBComposition**& composition, **int** axis, **unsigned int** index)

  Constructor specifying the particle $index$ and the $axis$ (0=x,1=y,2=z).
- **BBAlphaOperator**(const **BBAlphaOperator**& c)

  Copy constructor.
- virtual ∼**BBAlphaOperator**()

  Deconstructor.
- **unsigned int key**(**unsigned int** index) const

  Returns the component index (or key) on which the component $index$ acts.
- **BBAlphaOperator**& **operator**$*$**=**(const **BBAlphaOperator**& alpha)

  Multiplication assignment operator for two **BBAlphaOperator.** instances.
- **BBAlphaOperator**& **operator=**(const **BBAlphaOperator**& c)

  Assignment operator.

Private Attributes

- **int m_axis**

  Axial component index: x=0, y=1, z=2.
- **unsigned int m_particleIndex**

  Particle index.
- **vector**<**BBAlphaOperator**> **m_alpha**

  Coupling **BBAlphaOperator** instances.
- **BBComposition m_composition**

  BBComposition instance of the system for which this **BBAlphaOperator** is valid.
- **BBSigmaOperator m_sigma**

  BBSigmaOperator to form the α matrix.

Friends

- **BBRBasisFunction operator**$*$(const **BBAlphaOperator**& alpha, const **BBRBasisFunction**& function)

  Multiplication operator for an **BBAlphaOperator** instance $alpha$ and an **BBRBasisFunction** instance $function$. Returns a spin-transformed instance of **BBRBasisFunction**.
- **BBAlphaOperator operator**$*$(const **BBAlphaOperator**& alphal, const **BBAlphaOperator**& alphar)

  Multiplication operator for two **BBAlphaOperator** instances. $alphal$ is the left term and $alphar$ is the right term in the multiplication.

C.5 BBAntisymmetrizer Class Reference

BBAntisymmetrizer is a class which can generate permutation matrices needed for imposing the correct spin symmetry onto a basis function. It can be used to cycle through the permutation matrices in lexical order.

`#include <BBAntisymmetrizer.h>`

Public Member Functions

- **BBAntisymmetrizer**()

  Standard constructor.
- **BBAntisymmetrizer** (**BBComposition**& composition)

  Constructor taking the $composition$ of the system.
- $\sim$**BBAntisymmetrizer**()

  Deconstructor.
- **bool next**(**BBComposition**& composition)

  Loads next matrix and returns true if a cycle was completed. It requires the $composition$ of the system.
- **double sign**(**BBComposition**& composition)

  Returns the sign of the permutation. This is always positive for bosons, but each odd permutation of fermions changes the total sign.
- **void matrix**(**BBComposition**& composition)

  Determines the new matrix. Requires the $composition$ of the system.
- **Mat**<**int**>& **matrix**()

  Returns the current permutation matrix.

- **unsigned int permutations**()

  Returns the number of total permutations. This is equal to the number steps in a full cycle.
- **void reset**(**BBComposition**& composition)

  Resets the matrix to the identity matrix. This will restart the cycle from the beginning.

Private Attributes

- **vector**<**BBPermutationMatrix**> **m_permutationMatrices**

  Stores the permutations matrices for the particle types.
- **Mat**<**int**> **m_matrix**

  Current permutation matrix.
- **unsigned int m_dimension**

  The dimension of the permutation matrix.
- **unsigned int m_permutations**

  Number of total permutations.

C.6 BBCache Class Reference

BBCache is a class which stores precalculated values for easy access. The files are read in form a YAML file and can also be stored as a YAML file using the **BBDataFile** class. The number of keys, i.e., the dimension of the data array can vary and is not limited by the implementation. For example, precalculated binomial coefficients can be stored as a two-dimensional array. The number of keys is then two. The data itself is stored in a one-dimensional array. Thus, the key are internally transformed from an n-dimensional key to a single values key corresponding to the internal data storage.

`#include <BBCache.h>`

Public Member Functions

- **BBCache**()

  Standard constructor.
- **BBCache**(**YAML::Node** node, **string**& filename)

  Constructor reading values from YAML structure $node$. The $filename$ will be used for error messages.

- **BBCache**(const **BBCache**& c)

  Copy constructor.

- virtual ∼**BBCache**()

  Deconstructor.

- **void data**(**YAML::Node**& node, **string**& filename)

  Loading the YAML formatted data from $node$ The $filename$ will be used for error messages.

- **unsigned int dimension**()

  Returns the number of keys.

- **double**& **at**(**unsigned int** i[ ])

  Returns a value from a multi-dimensional key i.

- **double**& **at**(**unsigned int** i)

  Returns a value from a one-dimensional key i.

- **void keys**(**YAML::Node** node)

  Loads the number of legal values for each key element from the original YAML structure $node$.

- **double**∗ **load**(**YAML::Node** node, **unsigned int** &N)

  Parses a YAML structure $node$ and returns all the data. The number of entries will be stored in N.

- **BBCache**& **operator=**(const **BBCache**& c)

  Assignment operator.

Private Attributes

- **vector**<**unsigned int**> **m_keys**

  Range of different key elements.

- **unsigned int m_size**

  Number of stored values.

- **double**∗ **m_data**

  Pointer storing values in an allocated array.

- **string m_fileName**

  File name for error messages.

C.7 BBComposition Class Reference

BBComposition stores the parameters of a calculation. This includes particle information, physical parameters, sampling parameters and characteristic quantum numbers. Particle information is stored as a list of types and a list of according numbers, rather than a list of individual particles. The spin quantum numbers for non-relativistic calculations are stored accordingly. The angular momentum quantum number denominators J and M_J store the orbital angular momentum for non-relativistic calculations and the total angular momentum for relativistic calculations. The speed of light is also stored and set to 137.0359895 a.u. as a default but can be changed to a different value. Sampling parameters involve the standard deviations and means for both the matrices containing the α values in Eq. (2.42) and u global vectors by particle type. The c_A parameter in Eq. (2.42) will be set to 1 by default but can be changed. Also factorials are stored as an instance of **BBCache**. The according instance of **BBCache** has to be generated and stored after initialization.

C.7.1 Description

`#include <BBComposition.h>`

Public Member Functions

- **BBComposition**()

  Standard constructor.
- **BBComposition**(const **BBComposition**& **c**)

  Copy constructor.
- **BBComposition**(**YAML::Node** node, **YAML::Node** particles)

  Constructor loading YAML input from $node$ and YAML resource file $particles$ containing particle parameters.
- virtual ∼**BBComposition**()

  Deconstructor.
- **unsigned int size**() const

  Returns the number of particles.
- **unsigned int size**(**unsigned int** i)

  Returns the number of particles of type i.
- **unsigned int types**()

  Returns the number of particle types.

- **vector<BBParticle>& particles**()

  Returns a reference to the particle vector.
- **BBParticleType type**(**unsigned int** i)

  Returns the i^{th} type of the stored particles.
- **double mass**()

  Returns the total mass of all particles.
- **double mass**(**unsigned int** i)

  Returns the mass of particle i.
- **int charge**(**unsigned int** i)

  Returns the charge of particle i.
- **double& charge**()

  Returns a reference to the value of the charge of the central Coulomb potential.
- **int spin**(**unsigned int** i)

  Returns the value of the spin of particle i.
- **int& s**(**unsigned int** i)

  Returns the value of the S quantum number of particle type ensemble i.
- **int& ms**(**unsigned int** i)

  Returns the value of the M_S quantum number of particle group i.
- **int& j**()

  Returns the value of the J quantum number of the system.
- **int& mj**()

  Returns the value of the M_J quantum number of the system.
- **int& l**()

  Returns the value of the L quantum number of the system in case of LS coupling. Returns -1 otherwise.
- **int& s**()

  Returns the value of the S quantum number of the system in case of LS coupling. Returns -1 otherwise.
- **bool& bo**()

  Returns true if a BO calculation, false if pre-BO.
- **BBParticle& at**(**unsigned int** i)

  Returns a reference to the i^{th} particle.
- **unsigned int offset**(**unsigned int** i)

  Returns the first particle index of i^{th} particle type.

- **double factor**(**unsigned int** i, **unsigned int j**)

  Returns the mass factor for i^{th} and j^{th} mass factor $(m(i)m(j)/m_t^2)$.
- **void load**(**YAML::Node** node, **YAML::Node** particles)

  Loads system composition from a YAML structure $node$ and YAML resource file $particles$ containing particle parameters.
- **YAML::Node node**()

  Returns a YAML node containing all information of the current composition.
- **double**& **cA**()

  Returns a reference to the c_A factor.
- **double**& **sdA**(**int** i, **int** j)

  Returns a reference to the standard-deviation value for the α sampling-parameter for particles i and j.
- **mat**& **sdA**()

  Returns a reference to the standard-deviation matrix.
- **void sdA**(**double** a, **int** i, **int** j)

  Sets the standard-deviation value for the α sampling-parameter for particles i and j.
- **double**& **meanA**(**int** i, **int** j)

  Returns a reference to the mean value for the α sampling-parameter for particles i and j.
- **mat**& **meanA** ()

  Returns a reference to the mean value matrix.
- **void meanA** (**double** a, **int** i, **int** j)

  Sets the mean value for the α sampling-parameter for particles i and j.
- **double**& **sdU** (**int** i)

  Returns a reference to the standard-deviation value for the u sampling-parameter for particle i.
- **vec**& **sdU** ()

  Returns a reference to the standard-deviation vector.
- **double**& **meanU** (**int** i)

  Returns a reference to the mean value for the u sampling-parameter for particle i.
- **vec**& **meanU**()

  Returns a reference to the mean vector.
- **double**& **alphaMin**()

Returns a reference to minimal sample-value for the α parameters.

- **double**& **alphaMax**()

  Returns a reference to maximal sample-value for the α parameters.

- **void factorials**(**BBCache** cache)

  Stores the factorial values stored in $cache$.

- **double factorials** (**unsigned int** i)

  Returns the factorial for the value i.

- **void binomials** (**BBCache** cache)

  Stores the binomial coefficients stored in $cache$.

- **double binomials** (**unsigned int** i, **unsigned int** j)

  Returns the binomial coefficient for the pair of values (i, j).

- **double**& **c**()

  Returns a reference to the value of the speed of light.

- **void update**()

  Calculates the mass factors, total mass, total number of particles and the particle indices for direct particle access.

- **void calculateMassFactors**()

  Calculates the mass factors for generating $\boldsymbol{A}$ matrices from the α parameters.

- **void calculateTotalMass**()

  Calculates the total mass of the system.

- **void calculateNumberOfParticles**()

  Calculates the total number of particles of system.

- **void calculateParticleIndices**()

  Calculates the particle indices for direct particle access.

- **BBComposition**& **operator=**(const **BBComposition**& **c**)

  Assignment operator.

Private Attributes

- **int m_j**

  Total/Orbital angular momentum J/L quantum number.

- **int m_mj**

  Total/Orbital angular momentum M_J/M_L quantum number.

- **int m_totalL**

Spatial angular momentum quantum number L used for generating total angular momentum state J.

- **int m_totalS**

  Spin quantum number S used for generating total angular momentum state J.

- **vector<BBParticle> m_particles**

  Particle types contained in the system.

- **vector<int> m_particleNumbers**

  Number of each particle type.

- **vector<int> m_s**

  Vector for the S^2 quantum numbers in the non-relativistic theory.

- **vector<int> m_ms**

  Vector for the M_S quantum numbers in the non-relativistic theory.

- **double m_totalMass**

  Total mass.

- **unsigned int m_numberParticles**

  Total number of particles.

- **vector<unsigned int> m_particleIndex**

  Stores information for the direct access of i^{th} particle in m_particles.

- **double m_centralCharge**

  Nuclear charge for atomic BO calculations.

- **bool m_bo**

  True if it is a BO calculation, false if pre-BO.

- **double m_ca**

  c_A factor for the Gaussian $\boldsymbol{A}$ matrix

- **mat m_massFactors**

  Matrix containing the mass factors: $m(i)/m_t * m(j)/m_t$.

- **mat m_sdA**

  Standard-deviation values for the sampling of the α parameters.

- **mat m_meanA**

  Mean values for the sampling of the α parameters.

- **vec m_sdU**

  Standard-deviation values for the sampling of the u parameters.

- **vec m_meanU**

Mean values for the sampling of the u parameters.

- **BBCache m_factorials**

  Storing precalculated factorial values.
- **BBCache m_binomials**

  Storing binomial precalculated coefficients values.
- **double m_alphaMin**

  Minimal value for the sampling of the α parameters.
- **double m_alphaMax**

  Maximal value for the sampling of the α parameters.
- **double m_speedOfLight**

  Stores speed of light.

C.8 BBDataFile Class Reference

A class for YAML file i/o. Only reads/writes from/to disk. It does not interpret or parse data. Only YAML type files are supported.

`#include <BBDataFile.h>`

Public Member Functions

- **BBDataFile**()

  Standard constructor.
- virtual ~**BBDataFile**()

  Deconstructor.
- **BBDataFile**(const **BBDataFile**& c)

  Copy constructor.
- **BBDataFile**(**string** file, **BBDFMode** mode)

  Constructor taking filename and i/o mode.
- **void read**()

  Reads in YAML file.
- **void write**()

  Writes YAML file to disk.
- **void write**(**string** s)

  Writes string s to disk.
- **BBDFMode**& **mode**()

Returns i/o mode.

- **string& file**()

  Returns the file name.

- **YAML::Node node**()

  Returns the stored YAML node.

- **void load**(**YAML::Node** node)

  Loads YAML file from *node*.

- **void append**(**YAML::Node** node)

  Appends *node* to already stored YAML structure.

Private Attributes

- **string m_fileName**

  File name.

- **YAML::Node m_file**

  YAML file.

- **BBDFMode m_mode**

  i/o mode

C.9 BBEnumMap Class Reference

Interconverts particle/integral enums and strings. All functions are static.

C.9.1 Description

`#include <BBEnumMap.h>`

Public Member Functions

- **BBEnumMap**()

  Standard constructor.

- virtual ~**BBEnumMap**()

  Deconstructor.

Static Public Member Functions

- static **void initialise**()

  Initializes the maps storing enum and string associations.
- static **string toString**(**BBParticleType** p)

  Returns the name of the particle type p as a string.
- static **string toString**(**BBIntegralType** p)

  Returns the name of the integral type p as a string.
- static **string toString**(**BBKineticBalance** p)

  Returns the name of the kinetic balance type p as a string.
- static **BBIntegralType toIntegral**(**string** s)

  Returns integral type parsed from string s.
- static **BBParticleType toParticle**(**string** s)

  Returns particle type parsed from string s.
- static **BBKineticBalance toKineticBalance**(**string** s)

  Returns kinetic balance type parsed from string s.

Static Private Attributes

- static **map**<**string, BBParticleType**> **m_particleToString**

  Map with particle strings as keys.
- static **map**<**BBParticleType, string**> **m_stringToParticle**

  Map with particle types as keys.
- static **map**<**string, BBIntegralType**> **m_integralToString**

  Map with integral strings as keys.
- static **map**<**BBIntegralType, string**> **m_stringToIntegral**

  Map with integral types as keys.
- static **map**<**string, BBKineticBalance**> **m_kineticBalanceToString**

  Map with integral strings as keys.
- static **map**<**BBKineticBalance, string**> **m_stringToKineticBalance**

  Map with integral types as keys.

C.10 BBErrorHandler Class Reference

A class for error handling.

C.10.1 Description

A class which can handle errors and warnings. In case of a severe error, the program will be automatically terminated after printing an error message.

`#include <BBErrorHandler.h>`

Public Member Functions

- **BBErrorHandler**()

  Standard constructor.
- virtual ∼**BBErrorHandler**()

  Deconstructor.
- **void checkErrorValue**(**BBError** e, **string** s1="", **string** s2="")

  Checks error values. Two stings $s1$ and $s2$ can be used for adding additional information.

C.11 BBGaussian Class Reference

BBGaussian represents the radial Gaussian part of an ECG. It contains the α parameters and the correlation matrix $\boldsymbol{A}$. It performs appropriate sampling of the α parameters. The sampling distinguishes between the BO and pre-BO descriptions through the **BBComposition** class. **BBComposition** also provides the sampling parameters. The factor for the quasi-normalization is stored numerically for the integral evaluation using CECG.

`#include <BBGaussian.h>`

Public Member Functions

- **BBGaussian**()

  Standard constructor.
- **BBGaussian**(const **BBGaussian**& c)

  Copy constructor.
- **BBGaussian**(**BBComposition**& composition)

  Constructor taking the $composition$ of the system.
- **BBGaussian**(**YAML::Node** node, **BBComposition**& composition)

  Constructor loading values from YAML structure $node$.
- virtual ∼**BBGaussian**()

Deconstructor.

- **mat**& **a**()

  Returns a reference to the correlation matrix $\boldsymbol{A}$.

- **void a**(**BBComposition**& composition)

  Generates the correlation matrix $\boldsymbol{A}$ from stored α values.

- **mat**& **alpha**()

  Returns a reference to the matrix containing the α values.

- **void alpha**(**BBComposition**& composition)

  Samples a set of α values according to the sampling parameters stored in $composition$.

- **double**& **norm**()

  Returns a reference to the numerical quasinormalization factor.

- **void norm**(**int** l, **int** k, **vec** u)

  Calculates the normalization factor for non-relativistic ECGs using the parameters of a GVR: l is the L quantum number, k is the K value and u is the global vector.

- **void symmetry**(**Mat**<**int**>& p)

  Transforms $\boldsymbol{A}$ and the α parameters according to the Permutation matrix p.

- **YAML::Node node**(**BBDataFormat** format, **bool** minimal=true)

  Returns a YAML structure of the instance. All information will be saved if $minimal$ is set to false. Otherwise only the α parameters will be saved. The $\boldsymbol{A}$ matrix can be reconstructed from the α parameters. Numerical values will be stored as text if $format$ is set to **BBAscii** and as b64 represented binary if $format$ is set to **BBBinary**.

- **void load**(**YAML::Node** node, **BBComposition**& composition)

  Loads a YAML structure $node$ of the instance.

- **BBGaussian**& **operator=**(const **BBGaussian**& c)

  Assignment operator.

Private Attributes

- **mat m_a**

  Correlation matrix $\boldsymbol{A}$.

- **mat m_alpha**

  Matrix containing the α parameters.

- **double m_norm**

  Numerical value for the quasi-normalization.

C.12 BBIntegral Class Reference

BBIntegral is responsible for the evaluation of the integrals. Integrals can be evaluated at different levels such as at ECG and CECG level or at basis function level. The type of integral is specified in the constructor and does not need to be specified during the evaluation process.

C.12.1 Description

`#include <BBIntegral.h>`

Public Member Functions

- **BBIntegral**()

  Standard constructor.
- **BBIntegral**(const **BBIntegral**& c)

  Copy constructor.
- **BBIntegral**(**BBComposition**& composition, **BBIntegralType** type)

  Constructor defining the integral $type$. Loads externally stored precalculated values.
- virtual ∼**BBIntegral**()

  Deconstructor.
- **BBIntegralType type**()

  Returns the type of the integral.
- **complex**<**double**> **Evaluate**(**BBComposition**& composition, **BBGaussian**& gbra, **BBGaussian**& gket, **BBPolynomial**& pbra, **BBPolynomial**& pket, **Mat**<**int**> P, **unsigned int** p1=-1, **unsigned int** p2=-1)

  Evaluates the integral at ECG and CECG level. $gbra$ and $bket$ are the **BBGaussian** of the bra and the ket. The $pbra$ and the $pket$ are the **BBPolynomial** of the bra and the ket. P is a permutation matrix, and $p1$ and $p2$ are particle indices which are required by one and two particle operators.
- **double Evaluate**(**BBComposition**& composition, **BBNRBasisFunction**& bra, **BBNRBasisFunction**& ket, **unsigned int** p1=0)

Evaluates the integral at the **BBNRBasisFunction** level for the non-relativistic case. bra and ket are the according **BBNRBasisFunction** instances. $p1$ is required by one particle operators.

- **double Evaluate**(**BBComposition**& composition, **BBRBasisFunction**& bra, **BBRBasisFunction**& ket, **unsigned int** p1=0)

  Evaluates the integral at the **BBRBasisFunction** level for the relativistic case. bra and ket are the according **BBRBasisFunction** instances. $p1$ is required by one particle operators.

- **double Evaluate**(**BBComposition**& composition, **BBNRBasisFunction**& bra, **BBNRBasisFunction**& ket, **Mat**<**int**> p, **unsigned int** p1=0, **unsigned int** p2=0)

  Evaluates the integral at the **BBNRBasisFunction** level for the non-relativistic case or at component level in the relativistic case for a specific permutation P. bra and ket are the according **BBNRBasisFunction** instances. $p1$ and $p2$ are particle indices which are required by one and two particle operators.

- **double EvaluateSigmaP**(**BBComposition**& composition, **BBRBasisFunction**& bra, **BBRBasisFunction**& ket, **BBSigmaPOperator**& o, **unsigned int** p1)

  Evaluates the integral for the $\sigma \cdot p$ operator at **BBRBasisFunction** level. bra and ket are the according **BBRBasisFunction** instances. o is the **BBSigmaPOperator** instance for particle $p1$.

- **double EvaluateGaunt**(**BBComposition**& composition, **BBRBasisFunction**& bra, **BBRBasisFunction**& ket)

  Evaluates the Gaunt integral at **BBRBasisFunction** level. bra and ket are the according **BBRBasisFunction** instances.

- **double EvaluateGaunt**(**BBComposition**& composition, **BBNRBasisFunction**& bra, **BBNRBasisFunction**& ket, **Mat**<**int**> p, **unsigned int** p1, **unsigned int** p2, **unsigned int** axis)

  Evaluates the Gaunt integral at **BBNRBasisFunction** level for two particles along a certain axis and for a specific permutation bra and ket are the according **BBRBasisFunction** instances. p is the permutation matrix and $p1$ and $p2$ are the particle indices. $axis$ is 0 for the x-axis, 1 for the y-axis and 2 for the z-axis.

- **double EvaluateBreitR3**(**BBComposition**& composition, **BBRBasisFunction**& bra, **BBRBasisFunction**& ket)

  Evaluates the Breit integral at **BBRBasisFunction** level. bra and ket are the according **BBRBasisFunction** instances.

- **double COverlap**(**Mat**<**int**>& indices, **mat**& inverse, **int** size, **double** s0)

Evaluates the recursive CECG Overlap integral. $indices$ contains the power indices of the current term in the polynomial. $inverse$ contains the inverse of the bra and ket correlation matrix $\boldsymbol{A}_I + \boldsymbol{A}_J$. $size$ contains the number of particles. $s0$ is the integral value at the end of the recursion.

- **double CCoulomb**(**Mat**<**int**>& indices, **mat**& inverse, **int** size, **double** s0, **Col**<**int**>& n, **double**& Z, **unsigned int**& p1, **unsigned int**& p2)

  Evaluates the recursive CECG Coulomb integral. $indices$ contains the power indices of the current term in the polynomial. $inverse$ contains the inverse of the bra and ket correlation matrix $\boldsymbol{A}_I + \boldsymbol{A}_J$. $size$ contains the number of particles. $s0$ is the integral value at the end of the recursion. n is the auxiliary factor in the recursion. $p1$ and $p2$ are the indices of the two particles for which the interaction will be calculated. The integral will be evaluated according to the derivation scheme presented by Saito [267].

- **double CCoulomb2**(**Mat**<**int**>& indices, **mat**& inverse, **int** size, **double** s0, **double** m, **double**& rho, **unsigned int**& p1, **unsigned int**& p2)

  Evaluates the recursive CECG Coulomb and Central Potential integral. $indices$ contains the power indices of the current term in the polynomial. $inverse$ contains the inverse of the bra and ket correlation matrix $\boldsymbol{A}_I + \boldsymbol{A}_J$. $size$ contains the number of particles. s0 is the integral value at the end of the recursion. m is the auxiliary index and rho is the ρ factor in the recursion relation. p1 and p2 are the indices of the two particles for which the interaction will be calculated. The integral will be evaluated according to the derivation scheme presented by Shiozaki [270] (see Section 7.2).

- **double CLambda3**(**Mat**<**int**>& indices, **mat**& inverse, **int** size, **double** s0, **double** m, **double**& rho, **unsigned int**& p1, **unsigned int**& p2)

  Evaluates the recursive CECG Lambda3 inverse distance potential(r^{-3}) integral. $indices$ contains the power indices of the current term in the polynomial. $inverse$ contains the inverse of the bra and ket correlation matrix $\boldsymbol{A}_I + \boldsymbol{A}_J$. $size$ contains the number of particles. $s0$ is the integral value at the end of the recursion. m is the auxiliary index and rho is the ρ factor in the recursion relation. $p1$ and $p2$ are the indices of the two particles for which the interaction will be calculated. The integral will be evaluated according to the derivation scheme presented by Shiozaki [270] (see Section 7.2).

- **double CCentralPotential**(**Cube**<**int**>& indices, **mat**& inverse, **int** size, **double**$*$ s0, **Col**<**int**>& n, **double**& Z, **unsigned int**& p1)

  Evaluates the recursive CECG Central Potential integral. $indices$ contains the power indices of the current term in the polynomial. $inverse$ contains the inverse of the bra and ket correlation matrix $\boldsymbol{A}_I + \boldsymbol{A}_J$. $size$ contains

the number of particles. $s0$ is the integral value at the end of the recursion. n is the auxiliary factor in the recursion. $p1$ and $p2$ are the indices of the two particles for which the interaction will be calculated. The integral will be evaluated according to the derivation scheme presented by Saito [267].

- **double Wx**(**Col**<**int**>& n, **double**& Z, **int** M)

  Evaluates the auxilliary recursion relation for the index n according to Saito [267]. Z is the according factor in the recursion relation and M is the auxiliary index.

- **double Overlap**(**BBComposition**& composition, **BBGaussian**& gbra, **BBGaussian**& gket, **BBPolynomial**& pbra, **BBPolynomial**& pket, **Mat**<**int**> P)

  Evaluates the ECG/GVR integrals for the overlap. $gbra$ and $gket$ are the **BBGaussian** of the bra and the ket. $pbra$ and $pket$ are the **BBPolynomial** for the bra and the ket. P is the permutation matrix.

- **double Kinetic**(**BBComposition**& composition, **BBGaussian**& gbra, **BBGaussian**& gket, **BBPolynomial**& pbra, **BBPolynomial**& pket, **Mat**<**int**> P)

  Evaluates the ECG/GVR integrals for the kinetic energy. $gbra$ and $gket$ are the **BBGaussian** of the bra and the ket. $pbra$ and $pket$ are the **BBPolynomial** for the bra and the ket. P is the permutation matrix.

- **double Coulomb**(**BBComposition**& composition, **BBGaussian**& gbra, **BBGaussian**& gket, **BBPolynomial**& pbra, **BBPolynomial**& pket, **Mat**<**int**> P)

  Evaluates the ECG/GVR integrals for the Coulomb energy. $gbra$ and $gket$ are the **BBGaussian** of the bra and the ket. $pbra$ and $pket$ are the **BBPolynomial** for the bra and the ket. P is the permutation matrix.

- **double CentralPotential**(**BBComposition**& composition, **BBGaussian**& gbra, **BBGaussian**& gket, **BBPolynomial**& pbra, **BBPolynomial**& pket, **Mat**<**int**> P)

  Evaluates the ECG/GVR integrals for the central spherical Coulomb potential. $gbra$ and $gket$ are the **BBGaussian** of the bra and the ket. $pbra$ and $pket$ are the **BBPolynomial** for the bra and the ket. P is the permutation matrix.

- **double Momentum**(**BBComposition**& composition, **BBGaussian**& gbra, **BBGaussian**& gket, **BBPolynomial**& pbra, **BBPolynomial**& pket, **Mat**<**int**> P, **unsigned int** i)

  Evaluates the ECG/GVR integrals for the one-particle momentum. $gbra$ and $gket$ are the **BBGaussian** of the bra and the ket. $pbra$ and $pket$ are the

BBPolynomial for the bra and the ket. P is the permutation matrix. i is the particle index.

- **double Dipole**(**BBComposition**& composition, **BBGaussian**& gbra, **BBGaussian**& gket, **BBPolynomial**& pbra, **BBPolynomial**& pket, **Mat**<**int**> P, **unsigned int** i)

  Evaluates the ECG/GVR integrals for the transition dipole integral for one particle. $gbra$ and $gket$ are the **BBGaussian** of the bra and the ket. $pbra$ and $pket$ are the **BBPolynomial** for the bra and the ket. P is the permutation matrix. i is the particle index.

- **BBIntegral**& **operator=**(const **BBIntegral**& c)

  Assignment operator.

Private Attributes

- **BBIntegralType m_type**

  Integral type.

- **vector**<**BBCache**> **m_caches**

  Vector of **BBCache** storing the precalculated values.

- **vector**<**BBAntisymmetrizer**> **m_symmetry**

  Vector of **BBAntisymmetrizer**. Size will be defined by the number of OMP nodes.

C.13 BBNRBasis Class Reference

BBNRBasis is a non-relativistic basis set. It contains a list of **BBNRBasisFunction**. The class contains functions for calculating the matrix representation of the Hamiltonian and for solving the generalized eigenproblem. Before an initial calculation can be performed, the matrices have to be reset. Furthermore, it contains functions for the optimization of the parameter set. It requires an instance of **BBComposition** in order to specify the system which will be described by the basis set.

```
#include <BBNRBasis.h>
```

Public Member Functions

- **BBNRBasis**()

  Standard constructor.

- **BBNRBasis**(const **BBNRBasis**& c)

  Copy constructor.

- **BBNRBasis**(**BBComposition**& composition, **int** size, **int** j, **int** mj, **int** S, **int** L, **int** maxK, **bool** polynomials)

  Constructor using the composition to generate a non-relativistic basis for a total angular momentum state. The number of basis functions is defined by $size$. The total angular momentum state is defined by j and mj. The total spin state is defined by S. The largest value for K is set by $maxK$ and if $polynomials$ is set true, then the polynomial representation of the GVR will be generated.

- **BBNRBasis**(**BBComposition**& composition, **int** size, **vector**<**int**> S, **vector**<**int**> MS, **int** l, **int** ml, **int** maxK, **bool** polynomials)

  Constructor using the composition to generate a non-relativistic basis for a total spatial angular momentum state and elementary spin states. The number of basis functions is defined by $size$. The total spatial angular momentum state is defined by l and ml. The elementary spins states are stored in S and MS. The largest value for K is set by $maxK$ and if $polynomials$ is set true, then the polynomial representation of the GVR will be generated.

- **BBNRBasis**(**YAML::Node** node, **BBComposition**& composition, **bool** polynomials)

  Constructor loading YAML input from $node$. If $polynomials$ is true, then the polynomial representation of the GVR will be generated.

- virtual ∼**BBNRBasis**()

  Deconstructor.

- **BBComposition**& **composition**()

  Returns a reference to the composition of the basis set.

- **void reset**()

  Resets the matrices for the integral matrices.

- **void resize**(**BBComposition**& composition, **int** size, **int** j, **int** mj, **int** S, **int** L, **int** maxK, **bool** polynomials)

  Changes the size of the basis for total angular momentum basis functions. The new number of basis functions will be $size$. The angular momentum state is j and mj. S is the total spin state. $maxK$ is the largest sample value for K in the GVR. If $polynomials$ is true, then the polynomial representation of the GVR will be generated.

- **void resize**(**BBComposition**& composition, **int** size, **vector**<**int**> S, **vector**<**int**>MS, **int** l, **int** ml, **int** maxK, **bool** polynomials)

Changes the size of the basis for total spatial angular momentum basis functions and elementary spin states. The new number of basis functions will be $size$. the angular momentum state is j and mj. The elementary spins states are stored in S and MS. $maxK$ is the largest sample value for K in the GVR. If $polynomials$ is true, then the polynomial representation of the GVR will be generated.

- **unsigned int size**()

  Returns the number of basis functions.

- **BBNRBasisFunction**& **at**(**unsigned int** i)

  Returns a reference to the i^{th} basis function.

- **vector**<**BBIntegral**>& **potentials**()

  Returns a reference to the vector containing the potential integrals.

- **void potentials**(**unsigned int** i, **unsigned int** j=0)

  Calculates the integrals for the potential operators. The i^{th} column and row of the integrals matrices are calculated starting from the j^{th} element. By default the whole column and row are calculated.

- **mat potentialMat**(**unsigned int** i)

  Returns a the matrix of the i^{th} potential in the Hamiltonian.

- **BBIntegral**& **metric**()

  Returns a reference to the overlap integral.

- **void metric**(**unsigned int** i, **unsigned int** j=0)

  Calculates the integrals for the overlap. The i^{th} column and row of the integrals matrices are calculated starting from the j^{th} element. By default the whole column and row are calculated.

- **mat metricMat**()

  Returns the overlap matrix.

- **BBIntegral**& **kinetic**()

  Returns a reference to the kinetic-energy integral.

- **void kinetic**(**unsigned int** i, **unsigned int** j=0)

  Calculates the integrals for the kinetic energy. The i^{th} column and row of the integrals matrices are calculated starting from the j^{th} element. By default the whole column and row are calculated.

- **mat kineticMat**()

  Returns the kinetic-energy matrix.

- **void loewdin**(**double** t=1.0e-12)

  Performs the Löwdin ortho-normalization. The threshold for selecting ortho-normal basis functions is t. It is by default 1e-12.

- **mat loewdinMat**()

  Returns the matrix representation of the Löwdin ortho-normalization transformation.

- **double evaluate**(**BBIntegral** integral, **int** v, **unsigned int** p1=0, **unsigned int** p2=0)

  Evaluates the expectation values for the operator specified by integral. $p1$ and $p2$ specify the particles for one- and two-particle operators.

- **void calculate**()

  Calculates all integral matrices.

- **void update**(**unsigned int** i)

  Updates column and row i of all integral matrices.

- **void solve**(**double** t=1.0e-12)

  Solves the generalized eigenproblem. It performs a Löwdin ortho-normalization with the threshold t. it is 1e-12 by default.

- **cx_vec solveCCR**(**double** angle, **double** t=1.0e-12)

  Solves the generalized eigenproblem for the Complex Coordinate Rotation method using the current integral matrices. The rotation is defined by $angle$. It performs a Löwdin ortho-normalization with the threshold t. It is 1e-12 by default.

- **cx_vec solveCAP**(**double** angle, **double** t=1.0e-12)

  Solves the generalized eigenproblem for the Complex Absorption Potential method using the current integral matrices. The rotation is defined by $angle$. It performs a Löwdin ortho-normalization with the threshold t. It is 1e-12 by default.

- **mat states**()

  Returns the eigenvectors of the solution of the generalized eigenproblem.

- **vec energies**()

  Returns the energies obtained from the solution of the generalized eigenproblem.

- **void sample**(**int** rounds, **double** t, **unsigned int** k, **bool** polynomials, **int** v)

  Optimizes the basis set by sampling of the correlation matrices $\boldsymbol{A}$. The number of sampling steps is defined by $rounds$. t is the threshold for the Löwdin ortho-normalization. k is the largest value for K in the GVR. The polynomial representation of the GVR will be generated if $polynomials$ is true. v is the vibrational quantum number.

- **void sampleU**(**int** rounds, **double** t, **int** v)

Optimizes the basis set by sampling of the global vector $\boldsymbol{u}$. The number of sampling steps is defined by $rounds$. t is the threshold for the Löwdin ortho-normalization. k is the largest value for K in the GVR. v is the vibrational quantum number.

- **void parameters**(**unsigned int** i, **unsigned int** j)

  Performs a statistical analysis of the α parameters in $\boldsymbol{A}$ for types i and j and changes the sampling parameters accordingly.

- **void parametersU**(**unsigned int** i)

  Performs a statistical analysis of the parameters in $\boldsymbol{u}$ for type i and changes the sampling parameters accordingly.

- **void parameters**()

  Performs a statistical analysis of the α parameters in $\boldsymbol{A}$ for all pairs of types and changes the sampling parameters accordingly.

- **void parametersU**()

  Performs a statistical analysis of the parameters in $\boldsymbol{u}$ for types and changes the sampling parameters accordingly.

- **void randomStepUniform**(**unsigned int** steps, **double** t, **double** range, **int** v)

  Optimizes the basis by performing random steps of the α parameters. The number of sampling steps is defined by $rounds$. t is the threshold for the Löwdin ortho-normalization. $range$ specifies the maximal step size. k is the largest value for K in the GVR. v is the vibrational quantum number.

- **mat lengthTransitionP**(**BBNRBasis**& final)

  Calculates the integral matrix for the length representation of the transition dipole operator $\hat{\boldsymbol{\mu}}_{+}^{(l)}$. $final$ specifies the final state of the transition.

- **mat lengthTransitionM**(**BBNRBasis**& final)

  Calculates the integral matrix for the length representation of the transition dipole operator $\hat{\boldsymbol{\mu}}_{-}^{(l)}$. $final$ specifies the final state of the transition.

- **mat lengthTransitionZ**(**BBNRBasis**& final)

  Calculates the integral matrix for the length representation of the transition dipole operator $\hat{\boldsymbol{\mu}}_{z}^{(l)}$. $final$ specifies the final state of the transition.

- **mat velocityTransitionM**(**BBNRBasis**& final)

  Calculates the integral matrix for the velocity representation of the transition dipole operator $\hat{\boldsymbol{\mu}}_{-}^{(v)}$. $final$ specifies the final state of the transition.

- **mat velocityTransitionZ**(**BBNRBasis**& final)

  Calculates the integral matrix for the velocity representation of the transition dipole operator $\hat{\boldsymbol{\mu}}_{z}^{(l)}$. $final$ specifies the final state of the transition.

- **mat velocityTransitionP**(**BBNRBasis**& final)

  Calculates the integral matrix for the velocity representation of the transition dipole operator $\hat{\boldsymbol{\mu}}_{+}^{(l)}$. $final$ specifies the final state of the transition.
- **mat hamiltonian**()

  Returns the matrix representation of the Hamiltonian.
- **double virial**(**int** v)

  Returns the virial for the vibrational state v.
- **double metricError**()

  Returns the maximal error of the Löwdin ortho-normalization.
- **void load**(**YAML::Node** node, **BBComposition**& composition, **bool** polynomials)

  Loads the basis from $node$. It requires the composition of the system and the polynomial representation of the GVR will be generated if $polynomial$ is true.
- **YAML::Node node**(**BBDataFormat** format, **bool** minial=true)

  Returns a YAML representation of the basis. $format$ specifies if the numerical values are stored as plain text ASCII of Binary. If $minimal$ is set to false, all information will be stored. $minimal$ is by default true.
- **void publication**(**YAML::Emitter**& node)

  Stores a YAML representation of the basis suited for publication in $node$.
- **void analyze**(**int** v)

  Prints the energy contributions from the different terms of the Hamiltonian to the total energy for state v.
- **void writeIntegrals**(**string** fileName)

  Writes integrals to file $fileName$.
- **void loadIntegrals**(**string** fileName)

  Loads integrals from file fileName.
- **BBNRBasis**& **operator=**(const **BBNRBasis**& c)

  Assignment operator.

Private Attributes

- **vector**<**BBNRBasisFunction**> **m_basisFunctions**

  Vector storing the basis functions.
- **vector**<**BBIntegral**> **m_potentialIntegrals**

  Vector storing the potential integrals of the Hamiltonian.

- **BBIntegral m_overlapIntegral**

  Overlap integral.
- **BBIntegral m_kineticIntegral**

  Kinetic-energy integral.
- **vector<mat> m_potentials**

  Vector of potential-energy integral matrices.
- **mat m_potential**

  Total potential-energy matrix.
- **mat m_metric**

  Integral matrix of the overlap.
- **mat m_kinetic**

  Integral matrix of the kinetic energy.
- **BBComposition m_composition**

  Composition of the system.
- **mat m_state**

  Eigenvectors of the generalized eigenproblem.
- **vec m_energies**

  Energies of the generalized eigenproblem.
- **mat m_loewdin**

  Löwdin ortho-normalization matrix.

C.14 BBNRBasisFunction Class Reference

BBNRBasisFunction represents a non-relativistic ECG with the GVR or a CECG. The Gaussian part is represented by a **BBGaussian** instance and the angular momentum part is represented by a list of spin states and spatial angular momentum state which is described by an instance of **BBPolynomial**. Each spin state is associated with the according spatial angular momentum part. If the basis function is an eigenfunction of the total spatial angular momentum operator, then the basis function will contains only one spin state/**BBPolynomial** pair. If the basis function is an eigenfunction of the total angular momentum operator, then it can contain several spin state/**BBPolynomial** pairs depending on the Clebsh-Gordan expansion.

```
#include <BBNRBasisFunction.h>
```

Public Member Functions

- **BBNRBasisFunction**()

  Standard constructor.

- ∼**BBNRBasisFunction**()

  Deconstructor.

- **BBNRBasisFunction**(const **BBNRBasisFunction**& c)

  Copy constructor.

- **BBNRBasisFunction**(**BBComposition**& composition, **int** j, **int** mj, **int** S, **int** L, **int** maxK, **bool** polynomials)

  Constructor using the composition to generate a non-relativistic basis function for a total angular momentum state. The total angular momentum state is defined by j and mj. The total spin state is defined by S and the total spatial angular momentum is defined by L. The angular part is then generated according to the LS coupling scheme. The largest value for K is set by $maxK$ and if $polynomials$ is set true, then the polynomial representation of the GVR will be generated.

- **BBNRBasisFunction**(**BBComposition**& composition, **vector**<**int**> S, **vector**<**int**> MS, **int** l, **int** ml, **int** maxK, **bool** polynomials)

  Constructor using the composition to generate a non-relativistic basis function for a total spatial angular momentum state and elementary spin states. The total spatial angular momentum state is defined by l and ml. The elementary spins states are stored in S and MS. The largest value for K is set by $maxK$ and if $polynomials$ is set true, then the polynomial representation of the GVR will be generated.

- **BBNRBasisFunction**(**YAML::Node** node, **BBComposition**& composition, **bool** polynomials)

  Constructor loading YAML input from $node$. If $polynomials$ is true, then the polynomial representation of the GVR will be generated.

- **BBGaussian**& **gaussian**()

  Returns a reference to the Gaussian.

- **BBPolynomial**& **polynomial**(**unsigned int** i)

  Returns a reference to the i^{th} term in the polynomial representation of the GVR.

- **vector**<**BBPolynomial**>& **polynomials**()

  Returns a reference to the vector of polynomials.

- **Col**<**int**> **spinPattern**(**BBComposition**& composition, **Mat**<**int**> p, **unsigned int** i, **unsigned int** j=0)

Returns a the spin pattern for a permutation p of element i of spin state j. The spin pattern is a vector of integers, where the each element represents a spin of a single particle and the value of the element represent its state.

- **complex**<**double**>& **factor**(**unsigned int** i, **unsigned int** j=0)

  Returns a reference to the factor of a spin state at position i for the spin state j.

- **complex**<**double**>& **factor**(**BBComposition**& composition, **Col**<**int**> pattern, **unsigned int** j=0)

  Returns a reference to the factor of a spin state at the position specified by the spin pattern stored in $pattern$ for the spin state j.

- **cx_vec factorVec**(**unsigned int** i)

  Returns the spin state i.

- **cx_vec factorVec**(**BBComposition** composition, **unsigned int** i, **Mat**<**int**> P)

  Returns the spin state i for a permutation P.

- **vector**<**cx_vec**>& **factors**()

  Returns the vector of all spin states.

- **double**& **norm**()

  Returns the numerical quasi-normalization factor.

- **void normalize**()

  Calculates the numerical quasi-normalization factor.

- **void convertToTotal**(**BBComposition**& composition, **int** j, **int** mj, **int** l, **int** S, **bool** polynomials)

  Returns a reference to the composition of the basis set.

- **void convertToSpatial**(**BBComposition**& composition, **vector**<**int**> S, **vector**<**int**> MS, **int** l, **int** ml, **bool** polynomials)

  Converts the basis function to a total spatial angular momentum form. S and MS contain the elementary spin states and l and ml specify the total spatial angular momentum state. If $polynomial$ is true, then the polynomial representation of the GVR will be generated.

- **unsigned int size**()

  Returns the number of polynomials.

- **BBNRBasisFunction at**(**unsigned int** i)

  Returns a basis function only containing the i^{th} spin state and the according polynomial.

- **void push_back**(**BBPolynomial** polynomial, **cx_vec** spin)

Adds a pair of spin state and polynomial to the basis function.

- **void resample**(**BBComposition**& composition)

  Generates a new guess for $\boldsymbol{A}$.

- **void resampleU**(**BBComposition**& composition)

  Generates a new guess for $\boldsymbol{u}$.

- **void resampleK**(**BBComposition**& composition, **unsigned int** k, **bool** polynomials)

  Generates a new guess for K.

- **void symmetry**(**BBComposition**& composition, **Mat**<**int**>& p, **int** sign)

  Performs a particle permutation p and a $sign$ change on the basis function.

- **void correctSpinSign**()

  Reorders the signs and imaginary factors between a spin state and its polynomial such that the first element of the spin state is real and positive.

- **void contract**()

  Contracts redundant spin states and polynomials to shorter forms.

- **BBNRBasisFunction braket**(**BBComposition**& composition, **BBNRBasisFunction**& ket, **Mat**<**int**> p)

  Forms a new basis functions from the product of two basis functions bra and ket for a particle permutation p of the ket.

- **YAML::Node node**(**BBDataFormat** format, **bool** minimal=true)

  Returns a YAML representation of the basis function. $format$ specifies if the numerical values will be stored as plain text ASCII of Binary. If $minimal$ is set to false, all information will be stored. minimal is by default true.

- **void load**(**YAML::Node** node, **BBComposition**& composition, **bool** polynomials)

  Loads the basis from $node$. It requires the $composition$ of the system and the polynomial representation of the GVR will be generated if $polynomial$ is true.

- **BBNRBasisFunction**& **operator+=**(const **BBNRBasisFunction**& rhs)

  Assignment addition operator.

- **BBNRBasisFunction**& **operator-=**(const **BBNRBasisFunction**& rhs)

  Assignment subtraction operator.

- **BBNRBasisFunction**& **operator**$*$**=**(const **double** rhs)

  Assignment multiplication operator.

- **BBNRBasisFunction**& **operator/=**(const **double** rhs)

  Assignment division operator.

- **BBNRBasisFunction**& **operator***=(const **BBPolynomial** rhs)

  Assignment multiplication operator.
- const **BBNRBasisFunction operator+**(const **BBNRBasisFunction**& rhs) const

  Addition operator.
- const **BBNRBasisFunction operator-**(const **BBNRBasisFunction**& rhs) const

  Subtraction operator.
- const **BBNRBasisFunction operator***(const **double** rhs) const

  Multiplication operator.
- const **BBNRBasisFunction operator/**(const **double** rhs) const

  Division operator.
- **BBNRBasisFunction**& **operator=**(const **BBNRBasisFunction**& c)

  Assignment operator.

Private Attributes

- **BBGaussian m_gaussian**

  Gaussian part.
- **vector**<**BBPolynomial**> **m_polynomials**

  Polynomial part.
- **vector**<**cx_vec**> **m_spinStates**

  Vector of spin states.

Friends

- const **BBNRBasisFunction operator***(const **double** lhs, const **BBNRBasisFunction**& rhs)

  Multiplication operator.
- const **BBNRBasisFunction operator***(const **BBPolynomial**& lhs, const **BBNRBasisFunction**& rhs)

  Multiplication operator.
- const **BBNRBasisFunction operator/**(const **double** lhs, const **BBNRBasisFunction**& rhs)

  Division operator.

- **BBNRBasisFunction operator**$*$(const **BBSigmaPOperator**& lhs, const **BBNRBasisFunction**& rhs)

  Multiplication operator.
- **BBNRBasisFunction operator**$*$(const **BBSigmaOperator**& lhs, const **BBNRBasisFunction**& rhs)

  Multiplication operator.

C.15 BBNRSpinOperator Class Reference

BBNRSpinOperator is a class which contains functions for generating the matrix representation of spin operators and the vector eigenstates. The operators can be generated for any number of particles with any spin. The $\hat{S}_z$, $\hat{S}_+$ and $\hat{S}_-$ operators have simple matrix form and can thus be generated directly. All other operators, such as $\hat{S}_x$, $\hat{S}_y$ and $\hat{S}^2$, are generated from the initial three operators. The eigenvectors are generated recursively through angular momentum recoupling.

`#include <BBNRSpinOperator.h>`

Public Member Functions

- **BBNRSpinOperator**()

  Standard constructor.
- **BBNRSpinOperator**(const **BBNRSpinOperator**& c)

  Copy constructor.
- virtual ∼**BBNRSpinOperator**()

  Deconstructor.
- **Col**<**int**> **z**(**BBComposition**& composition)

  Returns the $\hat{S}_z$ operator for the system described by $composition$.
- **Mat**<**int**> **total**(**BBComposition**& composition)

  Returns the $\hat{S}^2$ operator for the system described by $composition$.
- **Mat**<**int**> **raising**(**BBComposition**& composition)

  Returns the $\hat{S}_+$ operator for the system described by $composition$.
- **Mat**<**int**> **lowering**(**BBComposition**& composition)

  Returns the $\hat{S}_-$ operator for the system described by $composition$.
- **Mat**<**int**> **y**(**BBComposition**& composition)

  Returns the $\hat{S}_y$ operator for the system described by $composition$.

- **Mat**<**int**> **x**(**BBComposition**& composition)

  Returns the $\hat{S}_x$ operator for the system described by $composition$.

- **unsigned int dimension**(**BBComposition**& composition)

  Returns the matrix dimension of the spin operators.

- **unsigned int dimension**(**vector**<**BBParticle**>& particles)

  Returns the matrix dimension of the spin operators for the vector $particles$.

- **vec state**(**int** S, **int** MS, **BBComposition**& composition)

  Returns the total spin state for a system described by $composition$ for the state S, M_S.

- **vec state**(**vector**<**int**> S, **vector**<**int**> MS, **BBComposition**& composition)

  Returns the spin state for a system described by $composition$ for the elementary spin states listed in S and M_S.

- **vec state**(**int** S, **int** MS, **BBComposition**& composition, **vector**<**BBParticle**>& particles)

  Returns the spin state for a list of particles for the elementary spin states listed in S and MS.

- **Mat**<**int**> **spinStructure**(**BBComposition**& composition)

  Calculates all spin patterns.

Private Member Functions

- **int maxS**(**vector**<**BBParticle**> particles)

  Calculates the largest possible value for S for a list of particles.

C.16 BBParticle Class Reference

BBParticle contains the physical parameters of a particle type. The parameters include the mass, charge, spin and a type specifier. The values for the parameters have to be provided and are not stored internally.

```
#include <BBParticle.h>
```

Public Member Functions

- **BBParticle**()

  Standard constructor.
- **BBParticle**(const **BBParticle**& c)

  Copy constructor.
- **BBParticle**(**BBParticleType** type, **int** spin, **int** mass, **int** charge)

  Constructor specifying the type, spin, mass and charge of the particle.
- **BBParticle**(**BBParticleType** type)

  Constructor using only the type. The other parameters are to be specified later.
- **BBParticle**(**YAML::Node** node)

  Constructor loading parameters from a YAML node.
- virtual ∼**BBParticle**()

  Deconstructor.
- **BBParticleType**& **type**()

  Returns the type of the particle.
- **int**& **spin**()

  Returns the spin of the particle.
- **double**& **mass**()

  Returns the mass of the particle.
- **int**& **charge**()

  Returns the charge of the particle.
- **void load**(**YAML::Node** node)

  Loads parameters from a YAML node.
- **BBParticle**& **operator=**(const **BBParticle**& c)

  Assignment Operator.

Private Attributes

- **BBParticleType m_type**

  Type identifyer of the particle.
- **int m_spin**

  Spin of the particle.
- **double m_mass**

Mass of the particle.

- **int m_charge**

  Charge of the particle.

Friends

- **bool operator==**(const **BBParticle**& a, const **BBParticle**& b)

  Equal to operator.
- **bool operator!=**(const **BBParticle**& a, const **BBParticle**& b)

  Not equal to operator.

C.17 BBPermutationMatrix Class Reference

BBPermutationMatrix is an implementation of a particle-permutation matrix. It contains one permutational state at each time. It is possible to cycle through all permutations of a set of particles. A cycle starts at the identity matrix and all further matrices are generated in lexical order. An instance of **BBPermutationMatrix** only requires the number of particles to be permutated.

`#include <BBPermutationMatrix.h>`

Public Member Functions

- **BBPermutationMatrix**()

  Standard constructor.
- **BBPermutationMatrix**(**unsigned int** n)

  Constructor for n particles.
- **BBPermutationMatrix**(const **BBPermutationMatrix**& c)

  Copy constructor.
- virtual ~**BBPermutationMatrix**()

  Deconstructor.
- **void next**()

  Loads next permutation.
- **Mat**<**int**>& **matrix**()

  Returns matrix of current permutations.
- **unsigned int dimension**()

  Returns dimension of permutation matrix.

- **int parity**()

  Returns the sign of the current permutation.

- **bool last**()

  Returns true if the former permutation was the last permutation.

- **void reset**(**unsigned int** n)

  Resets the permutation to the initial state for n particles.

- **BBPermutationMatrix**& **operator=**(const **BBPermutationMatrix**& c)

  Assignment operator.

Private Attributes

- **Mat**<**int**> **m_matrix**

  Permutation matrix.

- **vector**<**unsigned int**> **m_currentPermutation**

  Vector or permuted particle indices.

- **bool m_last**

  True if the former permutation was the last, false otherwise.

C.18 BBPolynomial Class Reference

BBPolynomial is an implementation of the angular part of an ECG. It can represent a GVR or Cartesian polynomials. The Cartesian polynomials are not automatically generated. If they are generated, the according the current GVR (see Eq. (2.24)) is formed. K is always assumed to be zero if the polynomial representation is generated. It contains routines which generate a guess for the global vector $\boldsymbol{u}$ and it distinguishes between BO and pre-BO calculations. In pre-BO calculations, c_u is zero, if not otherwise specified. If the polynomial form is generated, then it is possible to use the multiplication operators. The GVR is ignored for all operators except the assignment operator. Polynomials can be multiplied with each other and can be multiplied and divided by scalar factors. Polynomials can also be added and subtracted from each other.

```
#include <BBPolynomial.h>
```

Public Member Functions

- **BBPolynomial**()

  Standard constructor.
- **BBPolynomial**(const **BBPolynomial**& c)

  Copy constructor.
- **BBPolynomial**(**YAML::Node** node, **BBComposition**& composition, **bool** polynomials)

  Constructor loading YAML input. If $polynomials$ is true, then the polynomial representation of the GVR will be generated.
- **BBPolynomial**(**BBComposition**& composition, **int** l, **int** ml, **int** k, **bool** polynomials)

  Constructor using GVR parameters. l and ml are the spatial angular momentum quantum numbers and k is the K factor in the GVR. If $polynomials$ is true, then the polynomial representation of the GVR will be generated.
- virtual ∼**BBPolynomial**()

  Deconstructor.
- **void u**(**BBComposition**& composition)

  Samples a new global vector $\boldsymbol{u}$.
- **vec**& **u**()

  Returns a reference to the global vector u.
- **int**& **l**()

  Returns a reference to the L quantum number.
- **int**& **ml**()

  Returns a reference to the M_L quantum number.
- **int**& k()

  Returns a reference to the K factor in the GVR.
- **void polynomials**(**BBComposition**& composition)

  Generates polynomial representation of the GVR.
- **void indices**(**Mat**<**int**>& indices, **cx_vec**& prefactors, **int** k, **BBComposition**& composition, **bool** pref=true)

  Generates the polynomial terms recursively for the expanded form of a polynomial of type $(x_1 + ... + x_N)^k$. In $indices$ are the calculated exponents returned and in $prefactors$ are the calculated multiplicative factors of each polynomial term returned. k is the order of the polynomial. If $pref$ is set true, the factors of the entries of the global vectors will be included in the calculation. Otherwise the elements will be assumed to be 1.

- **Col**<**int**> **indices**(**unsigned int** c, **unsigned int** i)

  Returns exponentials of term c for axis i.
- **Mat**<**int**> **indices**(**unsigned int** i, **Mat**<**int**> P)

  Returns exponentials for axis i for the permutation P.
- **Mat**<**int**> **indices**(**unsigned int** i)

  Returns exponentials for axis i.
- **Cube**<**int**>& **indices**()

  Returns all exponentials. Each slice in the returned **Cube**<**int**> represents an axis and each row represents a term.
- **BBPolynomial conj**()

  Returns the complex conjugated polynomial.
- **complex**<**double**>& **factors**(**unsigned int** i)

  Returns a reference to the multiplicative prefactor of term i in the polynomial.
- **cx_vec**& **factors**()

  Returns a reference to all multiplicative prefactors in the polynomial.
- **int size**()

  Returns the number of terms in the polynomial.
- **void symmetry**(**Mat**<**int**>& p, **int** sign)

  Performs a particle permutation where p is the permutation matrix and $sign$ specifies the sign change on the polynomial.
- **void contract**()

  Contracts terms which appear multiple times in the polynomial.
- **YAML::Node node**(**BBDataFormat** format, **bool** minimal=true)

  Returns a YAML representation of the polynomial. If $minimal$ is set to false, all information will be stored. $minimal$ is by default true. Numerical values will be stored as text if $format$ is set to **BBAscii** and as b64 represented binary if $format$ is set to **BBBinary**.
- **void load**(**YAML::Node** node, **BBComposition**& composition)

  Loads the polynomial from $node$.
- **BBPolynomial**& **operator+=**(const **BBPolynomial**& c)

  Assignment addition operator.
- **BBPolynomial**& **operator-=**(const **BBPolynomial**& c)

  Assignment subtraction operator.
- **BBPolynomial**& **operator**$*$**=**(const **double**& c)

Assignment multiplication operator.

- **BBPolynomial**& **operator/=**(const **double**& c)

  Assignment division operator.

- **BBPolynomial**& **operator*=**(const **BBPolynomial**& c)

  Assignment multiplication operator.

- const **BBPolynomial operator+**(const **BBPolynomial**& c)

  Addition operator.

- const **BBPolynomial operator-**(const **BBPolynomial**& c)

  Subtraction operator.

- const **BBPolynomial operator***(const **double**& c)

  Multiplication operator.

- const **BBPolynomial operator/**(const **double**& c)

  Division operator.

- const **BBPolynomial operator***(const **BBPolynomial**& c) const

  Multiplication operator.

- **BBPolynomial**& **operator=**(const **BBPolynomial**& c)

  Assignment operator.

Private Attributes

- **vec m_u**

  Global vector.

- **int m_l**

  L quantum number

- **int m_ml**

  M_L quantum number

- **int m_k**

  K factor in the GVR.

- **Cube<int> m_indices**

  Exponentials ordered as: slices=terms, row=axis, column=particle.

- **cx_vec m_prefactors**

  Multiplicative prefactors for each term in the polynomial.

Friends

- const **BBNRBasisFunction operator**$*$(const **BBPolynomial**& lhs, const **BBNRBasisFunction**& rhs)

  Multiplication operator.

C.19 BBRBasis Class Reference

BBRBasis is a relativistic basis set. It contains a list of **BBRBasisFunctions**. The class contains functions for calculating the matrix representation of the Hamiltonian and solve the generalized eigenproblem. Before an initial calculation can be performed, the matrices have to be reset. Furthermore, it contains functions for the optimization of the parameter set. It requires an instance of **BBComposition** in order to specify the system which is described by the basis set.

`#include <BBRBasis.h>`

Public Member Functions

- **BBRBasis**()

  Standard constructor.
- **BBRBasis**(**BBRBasis**& c)

  Copy constructor.
- **BBRBasis**(**BBComposition**& composition, **int** size, **int** j, **int** mj, **int** S, **int** L, **int** maxK, **BBKineticBalance** balance)

  Constructor generating a relativistic basis for a total angular momentum state. $composition$ specifies the system which is represented by the basis. The number of basis functions is defined by $size$. The total angular momentum state is defined by j and mj. The total spin state is defined by S. The largest value for K is set by $maxK$ and if $polynomials$ is set true, then the polynomial representation of the GVR will be generated. $balance$ states which kind of kinetic-balance condition will be used.
- **BBRBasis**(**BBComposition**& **composition**, **int** size, **vector**<**int**> S, **vector**<**int**> MS, **int** l, **int** ml, **int** maxK, **BBKineticBalance** balance)

  Constructor generating a non-relativistic basis for a total spatial angular momentum state and elementary spin states. $composition$ specifies the system which is represented by the basis. The number of basis functions is defined by $size$. The total spatial angular momentum state is defined

by l and ml. The elementary spins states are stored in S and MS. The largest value for K is set by $maxK$ and if $polynomials$ is set true, then the polynomial representation of the GVR will be generated. $balance$ states which kind of kinetic-balance condition will be used.

- **BBRBasis**(**YAML::Node** node, **BBComposition**& composition)

  Constructor loading YAML input.

- virtual ∼**BBRBasis**()

  Deconstructor.

- **BBComposition**& **composition**()

  Returns a reference to the composition of the basis set.

- **void reset**()

  Resets the integral matrices.

- **unsigned int size**()

  Returns the number of basis functions.

- **BBRBasisFunction**& **at**(**unsigned int** i)

  Returns a reference to the i^{th} basis function.

- **vector**<**BBIntegral**>& **potentials**()

  Returns a reference to the vector containing the potential-energy integrals.

- **void potentials**(**unsigned int** i, **unsigned int** j=0)

  Calculates the integrals for the potential operators. The i^{th} column and row of the integrals matrices are calculated starting from the j^{th} element. By default the whole column and row are calculated.

- **mat potentialMat**(**unsigned int** i)

  Returns a the matrix of the i^{th} potential-energy operator in the Hamiltonian.

- **void metric**(**unsigned int** i, **unsigned int** j=0)

  Calculates the integrals for the overlap. The i^{th} column and row of the integrals matrices are calculated starting from the j^{th} element. By default the whole column and row are calculated.

- **mat metricMat**()

  Returns the overlap matrix.

- **void kinetic**(**unsigned int** i, **unsigned int** j=0)

  Calculates the integrals for the kinetic-energy. The i^{th} column and row of the integrals matrices are calculated starting from the j^{th} element. By default the whole column and row are calculated.

- **mat kineticMat**()

  Returns the kinetic-energy matrix.

- **void resting**(**unsigned int** i, **unsigned int** j=0)

  Calculates the integrals for the rest energy. The i^{th} column and row of the integrals matrices are calculated starting from the j^{th} element. By default the whole column and row are calculated.

- **mat restingMat**()

  Returns the rest-energy matrix.

- **void loewdin**(**double** t=1.0e-12)

  Performs the Löwdin ortho-normalization. The threshold for selecting ortho-normal basis functions is t. It is by default 1e-12.

- **mat loewdinMat**()

  Returns the matrix representation of the Löwdin ortho-normalization transformation.

- **void calculate**()

  Calculates all integral matrices.

- **void update**(**unsigned int** i)

  Updates column and row i of all integral matrices.

- **void normalize**()

  Normalizes the components of all basis functions.

- **bool normalize**(**unsigned int** i, **bool** del=true)

  Normalizes the components of the i^{th} basis function. If del is true, then the basis function will be resampled if numerical problems occur. del is by default true.

- **void solve**(**double** t=1.0e-12)

  Solves the generalized eigenproblem. It performs a Löwdin ortho-normalization with the threshold t. It is 1e-12 by default.

- **cx_vec solveCCR**(**double** angle, **double** t=1.0e-12)

  Solves the generalized eigenproblem for the Complex Coordinate Rotation method using the current integral matrices. The rotation is defined by $angle$. It performs a Löwdin ortho-normalization with the threshold t. It is 1e-12 by default.

- **cx_vec solveCAP**(**double** angle, **double** t=1.0e-12)

  Solves the generalized eigenproblem for the Complex Absorption Potential method using the current integral matrices. The rotation is defined by $angle$. It performs a Löwdin ortho-normalization with the threshold t. It is 1e-12 by default.

- **mat states**()

  Returns the eigenvectors of the solution of the generalized eigenproblem.

- **vec energies**()

  Returns the energies obtained from the solution of the generalized eigenproblem.

- **void sample**(**int** rounds, **double** t)

  Optimizes the basis set by sampling of the $\boldsymbol{A}$ correlation matrices. The number of sampling steps is defined by $rounds$. t is the threshold for the Löwdin ortho-normalization. k is the largest value for K in the GVR. The polynomial representation of the GVR will be generated if $polynomials$ is true. v is the vibrational quantum number.

- **void randomStepUniform**(**unsigned int** steps, **double** t, **double** range)

  Optimizes the basis by performing random steps of the α parameters. The number of sampling steps is defined by $rounds$. t is the threshold for the Löwdin ortho-normalization. $range$ specifies the maximal step size. k is the largest value for K in the GVR. v is the vibrational quantum number.

- **void parameters**()

  Performs a statistical analysis of the α parameters in $\boldsymbol{A}$ for all pairs of types and changes the sampling parameters accordingly.

- **void balanceTest**()

  Performs a test if the kinetic-balance condition is correctly imposed. It prints the results to screen.

- **void symmetryTest**()

  Performs a test if the particle exchange symmetry is correctly imposed. It prints the results to screen.

- **int groundstate**()

  Returns the index of the ground state in the energy vector.

- **void writeIntegrals**(**string** fileName)

  Writes integrals to the file with the name $fileName$.

- **void loadIntegrals**(**string** fileName)

  Loads integrals from the file with the name $fileName$.

- **void remove**(**unsigned int** i)

  Removes i^{th} basis function.

- **void load**(**YAML::Node** node, **BBComposition**& composition)

  Loads the basis from $node$.

- **YAML::Node node**(**BBDataFormat** format)

  Returns a YAML representation of the basis. Numerical values will be stored as text if $format$ is set to **BBAscii** and as b64 represented binary if $format$ is set to **BBBinary**.

- **double metricError**()

  Returns the maximal error of the Löwdin ortho-normalization.
- **mat hamiltonian**()

  Returns the matrix representation of the Hamiltonian.
- **void analyze**(**int** v)

  Prints the energy contributions from the different terms of the Hamiltonian to the total energy for state v.
- **BBRBasis**& **operator=**(const **BBRBasis**& c)

  Assignment operator.

Private Attributes

- **BBRBasisFunction**∗ **m_basisFunctions**

  Array storing the basis functions.
- **unsigned int m_size**

  Number of basis functions.
- **vector**<**BBIntegral**> **m_potentialIntegrals**

  Vector storing the potential-energy integrals of the Hamiltonian.
- **vector**<**BBSigmaPOperator**> **m_sigmaP**

  Vector storing **BBSigmaPOperator** for each particle.
- **BBIntegral m_overlapIntegral**

  Overlap integral.
- **BBIntegral m_SPIntegral**

  Kinetic-energy integral.
- **BBIntegral m_restingIntegral**

  Rest-energy integral.
- **vector**<**mat**> **m_potentials**

  Vector of potential-energy integral matrices.
- **mat m_potential**

  Total potential-energy matrix.
- **mat m_metric**

  Integral matrix of overlap.
- **mat m_kinetic**

  Integral matrix of kinetic-energy.
- **mat m_resting**

Integral matrix of rest energy.

- **BBComposition m_composition**

  Composition of the system.

- **mat m_state**

  Eigenvectors of the generalized eigenproblem.

- **vec m_energies**

  Energies of the generalized eigenproblem.

- **mat m_loewdin**

  Löwdin ortho-normalization matrix.

- **int m_groundstate**

  Index for the ground-state energy.

C.20 BBRBasisFunction Class Reference

BBRBasisFunction represents a relativistic many-fermion spinor. The spinor is partitioned according to Eq. (6.22). Each component is represented by an instance of **BBNRBasisFunction**. The components can be accesses by the index which is a single number specifying the vector element. Alternatively, it is possible to access a component through the Λ key according to Eq. (6.22). Here, l is represented by 0 and s is represented by 1. Thus the ls component of a two-fermion spinor is represented as a **Col**<**int**> with the elements (0,1). The quasi-normalization factor is generally not generated by itself. This is done by the **BBRBasis** class which allows for more flexibility. **BBRBasis-Function** contains the functions to impose the kinetic-balance condition on the non-relativistic limit. It supports both the explicitly correlated kinetic-balance condition and the one-fermion kinetic-balance condition.

`#include <BBRBasisFunction.h>`

Public Member Functions

- **BBRBasisFunction**()

  Standard constructor.

- **BBRBasisFunction**(const **BBRBasisFunction**& c)

  Copy constructor.

- **BBRBasisFunction**(**BBComposition**& composition, **vector**<**int**> S, **vector**<**int**> MS, **int** l, **int** ml, **int** k, **BBKineticBalance** balance)

Constructor generating a relativistic basis function for a total spatial angular momentum state and elementary spin states. $composition$ specifies which system is represented. The total spatial angular momentum state is defined by l and ml. The elementary spin states are stored in S and MS. k currently ignored. $balance$ specifies the kinetic-balance condition used.

- **BBRBasisFunction**(**BBComposition**& composition, **int** j, **int** mj, **int** S, **int** L, **int** k, **BBKineticBalance** balance)

  Constructor generating a relativistic basis function for a total angular momentum state. $composition$ specifies which system is represented. The total angular momentum state is defined by j and mj. The total spin state is defined by S and the total spatial angular momentum is defined by L. The angular part is then generated according to the LS coupling scheme. k is currently ignored. $balance$ specifies the kinetic-balance condition used.

- **BBRBasisFunction**(**YAML::Node** node, **BBComposition**& composition)

  Constructor loading YAML input.

- **BBRBasisFunction**(**YAML::Node** node, **BBComposition**& composition, **BBKineticBalance** balance)

  Constructor loading YAML input and specifying the kinetic-balance condition.

- **BBRBasisFunction**(**BBComposition**& composition, **BBNRBasisFunction** theta, **BBKineticBalance** balance)

  Constructor loading the non-relativistic limit YAML input and specifying the kinetic-balance condition.

- virtual ∼**BBRBasisFunction**()

  Deconstructor.

- **unsigned int key**(**Col**<**int**> i)

  Calculates the the component index from the Λ key stored in i. 0 represents l, 1 represents s.

- **BBNRBasisFunction**& **at**(**unsigned int** component, **unsigned int** symmetry=0)

  Returns a reference to the permutation specified by $symmetry$ of a component specified by $component$. By default, the original permutation is returned.

- **BBNRBasisFunction**& **at**(**Col**<**int**> component, **unsigned int** symmetry=0

  Returns a reference to the permutation specified by $symmetry$ of a component specified by the Λ key stored in $component$. By default, the original permutation is returned.

- **BBNRBasisFunction**& **theta**(**unsigned int** symmetry=0)

  Returns a reference to the non-relativistic limit for a permutation specified by $symmetry$. By default, the original permutation is returned.

- **BBPolynomial**& **polynomial**(**unsigned int** component, **unsigned int** polynomial, **unsigned int** symmetry=0)

  Returns a reference to the polynomial for a component specified by the index for a permutation specified by $symmetry$. By default, the original permutation is returned.

- **BBPolynomial**& **polynomial**(**Col**<**int**> component, **unsigned int** polynomial, **unsigned int** symmetry=0)

  Returns a reference to the polynomial for a component specified by the Λ key stored in $component$ for a permutation specified by $symmetry$. By default, the original permutation is returned.

- **BBGaussian**& **gaussian**(**unsigned int** symmetry=0)

  Returns a reference to the Gaussian. The permutation index is specified by $symmetry$. By default, the original permutation is returned.

- **complex**<**double**>& **factor**(**unsigned int** component, **unsigned int** spinor, **unsigned int** spinorelement, **unsigned int** symmetry=0)

  Returns a reference to the factor of a spin state at position $spinorelement$ for the spin-state spinor for a component specified by the index $component$ for a permutation specified by $symmetry$. By default, the original permutation is returned.

- **complex**<**double**>& **factor**(**BBComposition**& composition, **Col**<**int**> component, **Col**<**int**> spinor, **unsigned int** spinorelement, **unsigned int** symmetry=0)

  Returns a reference to the factor of a spin state at position $spinorelement$ for the spin-state spinor for a component specified by the Λ key $component$ for a permutation specified by $symmetry$. By default, the original permutation is returned.

- **void normalize**(**double** norm)

  Sets the numerical quasi-normalization factors to $norm$ for each component.

- **Col**<**int**> **pattern**(**BBComposition**& composition, **unsigned int** i)

  Returns the Λ key for the component i.

- **Col**<**int**> **spinPattern**(**BBComposition**& composition, **Mat**<**int**> p, **unsigned int** component, **unsigned int** spinor, **unsigned int** spinorelement)

  Returns a the spin pattern for a permutation matrix p of element $spinorelement$ of the spin-state spinor for a component specified by the index $component$.

The spin pattern is a vector of integers, where each element represents a spin of a single particle and the value of the element represent its state.

- **unsigned int size**(**unsigned int** i)

  Returns the number of spin states of component i.

- **unsigned int size**(**Col**<**int**> i)

  Returns the number of spin states of the component specified by the Λ key stored in i.

- **unsigned int size**()

  Returns the number of components.

- **unsigned int permutations**()

  Returns the number of permutations.

- **void resample**(**BBComposition**& composition)

  Generates a new guess for $\boldsymbol{A}$. kinetic-balance condition and the quasi-normalization are updated automatically.

- **void uniformStep**(**BBComposition**& composition, **double** range)

  Change $\boldsymbol{A}$ by performing a uniform step for the α parameters where the size is specified by $range$. The kinetic-balance condition and the quasi-normalization are updated automatically.

- **void balance**(**BBComposition**& composition, **BBKineticBalance** balance)

  Imposes the kinetic-balance condition specified by $balance$ onto the non-relativistic limit.

- **void balance**(**BBComposition**& composition)

  Imposes the kinetic-balance condition onto the non-relativistic limit.

- **void symmetry**(**BBComposition**& composition)

  Generates particle-permuted terms of the non-relativistic limit.

- **YAML::Node node**(**BBDataFormat** format, **bool** minimal=true)

  Returns a YAML representation of the basis function. Numerical values will be stored as text if $format$ is set to **BBAscii** and as b64 represented binary if $format$ is set to **BBBinary**.

- **void load**(**YAML::Node** node, **BBComposition**& composition)

  Loads the basis from $node$.

- **void loadNR**(**YAML::Node** node, **BBComposition**& composition, **BBKineticBalance** balance)

  Loads the basis from $node$ for a non-relativistic limit and a kinetic-balance condition specified by $balance$.

- **BBRBasisFunction**& **operator**$*$=(const **BBPolynomial**& rhs)

Assignment multiplication operator.

- **BBRBasisFunction**& **operator=**(const **BBRBasisFunction**& c)

  Assignment operator.

Private Attributes

- **vector**<**vector**<**BBNRBasisFunction**>> **m_components**

  Particle exchanged terms of components. First key is component key, second key is permutation key.

- **vector**<**BBNRBasisFunction**> **m_theta**

  Particle exchanged terms of non-relativistic limit.

- **BBKineticBalance m_balance**

  Kinetic-balance condition type.

- **vector**<**BBSigmaPOperator**> **m_sigmaP**

  Vector of $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operators for each particle.

- **BBAntisymmetrizer m_symmetry**

  (Anti-) symmetrization operator.

Friends

- **BBRBasisFunction operator**∗(const **BBAlphaOperator**& alpha, const **BBRBasisFunction**& function)

  Multiplication operator.

- **BBRBasisFunction operator**∗(const **BBPolynomial**& lhs, const **BBRBasisFunction**& function)

  Multiplication operator.

C.21 BBSigmaOperator Class Reference

BBSigmaOperator is an implementation of the σ_μ operator where $\mu \in \{x, y, z\}$ according to Eq. (3.2). It can act on an instance of **BBNRBasisFunction** and generate a spin-transformed instance.

`#include <BBSigmaOperator.h>`

Public Member Functions

- **BBSigmaOperator**()

  Standard constructor.

- **BBSigmaOperator**(**BBComposition**& composition, **int** axis, **unsigned int** index)

  Constructor specifying the particle index and the axis (0=x,1=y,2=z)

- **BBSigmaOperator**(const **BBSigmaOperator**& c)

  Copy constructor.

- virtual ∼**BBSigmaOperator**()

  Deconstructor.

- **cx_mat matrix**()

  Returns σ matrix.

- **BBSigmaOperator**& **operator*=**(const **BBSigmaOperator**& sigma)

  Assignment multiplication operator.

- **BBSigmaOperator**& **operator=**(const **BBSigmaOperator**& c)

  Assignement operator.

Private Attributes

- **int m_axis**

  Axis key. x=0, y=1, z=2.

- **unsigned int m_particleIndex**

  Particle index.

- **cx_mat m_matrix**

  σ matrix

- **BBComposition m_composition**

  Composition of the system.

- **vector**<**BBSigmaOperator**> **m_preceeding**

  Preceding σ operators. Used for chaining of several σ operators.

Friends

- **BBSigmaOperator operator**$*$(const **BBSigmaOperator**& lhs, const **BBSigmaOperator**& rhs)

  Multiplication operator.
- **BBNRBasisFunction operator**$*$(const **BBSigmaOperator**& lhs, const **BBNRBasisFunction**& rhs)

  Multiplication operator.

C.22 BBSigmaPOperator Class Reference

BBSigmaPOperator is an implementation of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator according to Eq. (3.2). It can act on an instance of **BBNRBasisFunction** to generate a transformed form if the instance of **BBNRBasisFunction** is represented in its CECG form.

`#include <BBSigmaPOperator.h>`

Public Member Functions

- **BBSigmaPOperator**()

  Standard constructor.
- **BBSigmaPOperator**(const **BBSigmaPOperator**& c)

  Copy constructor.
- **BBSigmaPOperator**(**BBComposition**& composition, **unsigned int** i)

  Constructor for the i^{th} particle in the system described by $composition$.
- virtual $\sim$**BBSigmaPOperator**()

  Deconstructor.
- **BBPolynomial act**(**BBPolynomial** polynomial, **int** axis, **int** particle, **double** prefactor) const

  Returns transformed polynomial. The transformation will be performed on the axial component specified by $axis$. $particle$ indicates the particle index of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator. $prefactor$ multiplies the transformed form by some factor.
- **BBPolynomial act**(**BBGaussian** polynomial, **int** axis, **int** particle, **double** prefactor) const

Returns transformed Gaussian. The transformation will be performed on the axial component specified by $axis$. $particle$ indicates the particle index of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator. $prefactor$ multiplies the transformed form by some factor.

- **BBNRBasisFunction act**(**BBNRBasisFunction** function, **int** axis, **int** particle, **double** prefactor) const

  Returns transformed non-relativistic basis function. The transformation will be performed on the axial component specified by $axis$ for all particles. $prefactor$ multiplies the transformed form by some factor.

- **BBNRBasisFunction act**(**BBNRBasisFunction** function, **int** axis, **double** prefactor) const

  Returns transformed non-relativistic basis function. The transformation will be performed on the axial component axis. particle states the particle of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator. prefactor multiplies the transformed form by some factor.

- **cx_mat matrix**(**unsigned int** i)

  Returns the $\boldsymbol{\sigma}$ matrix of the axis specified by i (x=0, y=1, z=2).

- **cx_cube**& **matrices**()

  Returns all $\boldsymbol{\sigma}$ matrices.

- **vec**& **particles**()

  Returns the linear combination of one-particle $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operators.

- **BBSigmaPOperator**& **operator**$*$=(const **BBSigmaPOperator**& rhs)

  Assignment multiplication operator.

- **BBSigmaPOperator operator**$*$(const **BBSigmaPOperator**& rhs)

  Multiplication operator.

- **BBSigmaPOperator**& **operator=**(const **BBSigmaPOperator**& c)

  Assignment operator.

Private Attributes

- **unsigned int m_index**

  Particle index.

- **vec m_particles**

  Linear combination of particle indices. Used for center-of-momentum frame.

- **unsigned int m_order**

  Order of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator. Facilitates evaluation of $(\boldsymbol{\sigma} \cdot \boldsymbol{p})^2$.

- **double m_prefactor**

Multiplicative prefactor.

- **cx_cube m_spinMatrices**

  Axial $\boldsymbol{\sigma}$ matrices.

- **vector<BBSigmaPOperator> m_preceeding**

  Preceeding $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operators. Used for chaining of several $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operators.

- **bool m_bo**

  True if a BO calculation is performed.

Friends

- **BBSigmaPOperator pow**(const **BBSigmaPOperator**& arg, **unsigned int** i)

  Calculates the power of a $\boldsymbol{\sigma} \cdot \boldsymbol{p}$ operator. i is the exponent.

- **BBNRBasisFunction operator***(const **BBSigmaPOperator**& lhs, const **BBNRBasisFunction**& rhs)

  Multiplication operator.

- **BBSigmaPOperator operator***(const **BBSigmaPOperator**& lhs, const **double**& rhs)

  Multiplication operator.

- **BBSigmaPOperator operator***(const **double**& lhs, const **BBSigmaPOperator**& rhs)

  Multiplication operator.

C.23 Enumerations

Enumerations

- enum **BBError** {
  BBAttributeWarning, **BBOptionsWarning**, **BBDataFileOpenWrite**, **BBNoError**,
  BBInputFileLoad, **BBParseComposition**, **BBParseOption**, **BBOptionsFatal**,
  BBAttributeFatal, **BBDataFileOpenRead**, **BBDataFileMode**, **BBSpinorMisfit**,
  BBNullAntisym, **BBCacheOOB**, **BBIntNaN**, **BBIntInf**,
  BBBasisFktAdd, **BBSpinorAdd**, **BBSpinorInsertOOB**, **BBSpinorAccessOOB**,
  BBNRSpinorQN, **BBParticleNA** }

List of error denominators.

- enum **BBParticleType** {
  E, **H1**, **H2**, **H3**,
  He3, **He4**, **Li6**, **Be9**,
  B10, **C12**, **Ps**, **Mu**,
  Pt195, **Ne19**, **Be13**, **C13**,
  N15, **O15**, **F19** }

  List of particle types.

- enum **BBDFMode** { **BBRead**, **BBWrite** }

  List of i/o modes.

- enum **BBIntegralType** {
  BBOverlap, **BBKinetic**, **BBCoulomb**, **BBMomentumZ**,
  BBMomentumP, **BBMomentumM**, **BBDipoleZ**, **BBDipoleP**,
  BBDipoleM, **BBCentralPotential**, **BBCOverlap**, **BBCKinetic**,
  BBCCoulomb, **BBCMomentumZ**, **BBCMomentumX**, **BBCMomentumY**,
  BBCRestEnergy, **BBCCentralPotential**, **BBCSigmaP**, **BBCGaunt**,
  BBCCoulomb2, **BBCBreit**, **BBCGaunt2**, **BBCCentralPotential2**,
  BBCOneParticleKinetic }

  List of integrals types.

- enum **BBDataFormat** { **BBBase64**, **BBAscii** }

  List of format types.

- enum **BBKineticBalance** { **BBOneParticle**, **BBExact** }

  List of kinetic balance types.

D

BlueBerry Manual

This is a short manual for the BlueBerry software which implements our framework for non-relativistic many-particle and relativistic many-1/2-fermion calculations. Calculations can be performed in a field-free environment as pre-BO calculations and with a spherical central Coulomb potential as BO calculations. In order to run BlueBerry, an input file is required. During the course of a calculation, the parameter set can be stored in different files. Also, a log file is created at the end of a calculation which saves all user defined parameters. This log file has the same structure as an input file and can be stored to perform the calculation again under identical starting conditions at a later time. All file in and output relies on the YAML file format [282] for standardization.

D.1 Input File

D.1.1 composition Block

The first part of the input file which has to be present, is the composition block. It primarily defines which particle types are involved in the calculation and how many particles of each type are present. Also, it defines the spin states of each particle type ensemble. Listing D.1 presents an example for the composition block of the input file for H_2^+.

particles Contains the information regarding the individual groups of particles. The label in the **type** block indicates the elementary particle or the

Listing D.1: An example for a composition block in a BlueBerry input file.

```
composition:
    particles:
      - type: H1
        number: 2
        s: 2
        ms: 0
      - type: E
        number: 1
        s: 1
        ms: 1
    angular momentum:
      j: 2
      mj: 0
    cA: 1
    born oppenheimer: false
```

isotope of the atomic nucleus. Additional information is stored externally and does not need to be specified. The currently supported particle types are: E (electron), H1 (proton), H2 (deuterium), H3 (triton), He3 (helium-3), He4 (helium-4), Ne19 (neon-19), Be13 (beryllium-13), C13 (carbon-13), N15 (nitrogen-15), O15 (oxygen-15), F19 (fluorine-19), Pt195 (platinum-195), Ps (positron) and Mu (muon). Note that, if the particle type represents an atom type, it will always represent the according nuclei. Thus Be13 does not represent the Beryllium-13 isotope atom but its nucleus.

In the case where the mass or the charge needs to be changed, it is not necessary to modify the external resources. It is possible to use the **mass** (mass in terms of the electron mass), **atomic mass** (atomic mass in terms of atomic mass units) and the **charge** blocks. In this example:

```
particles:
  - type: H1
    number: 1
    s: 1
    ms: 1
    mass: 1
```

the proton will be have charge $+1$ and mass 1 and therefore have the same properties as the positron.

Only one list entry should be present for each type of particle. The according particle exchange symmetry is only enforced on each block and the individual list entries in the **particles** block are not contracted. In this example:

```
    particles:
      - type: E
        number: 1
        s: 1
        ms: 1
      - type: E
        number: 1
        s: 1
        ms: 1
```

the particle exchange symmetry will not be imposed since the two electrons have individual blocks.

The **number** block states the number of particles of this type. If it is set to zero, the list entry will be skipped when BlueBerry parses the input file.

The spin quantum numbers are stated in the **s** and **ms** blocks. All spatial and spin angular momentum quantum numbers are multiplied by a factor of two in order to ensure integer (rather than half integer) numbers for the spin quantum numbers. Therefore, the system presented in Listing D.1 is in its ortho state regarding the spins of the protons (S_p=1).

angular momentum In the angular momentum block are the total (spatial) angular momentum quantum numbers specified through the **j** and **mj** blocks. Again, the angular momentum quantum numbers are multiplied by a factor of two. Thus, the system presented in Listing D.1 is in its first excited rotational state with $L = 1$. If the **l** and **s** quantum numbers are listed, instead of forming the total spatial angular momentum state, the total angular momentum state is constructed in terms of LS-coupling:

```
angular momentum:
    j: 2
    mj: 0
    l: 0
    s: 2
```

where the system is in a total angular momentum state with $J = 1$ and $M_J = 0$.

cA Defines the value of c_A in the parametrization Eq. (2.42). This allows the user to test that the calculations are translationally invariant by performing the calculations for a range of c_A values.

born oppenheimer Is either set to true or false and defines if the calculation will be a BO or a pre-BO calculation. This will affect the parametrization of the basis functions and decide whether translational invariance will be imposed or not. If this block is set to true, the parametrization of $\boldsymbol{A}$ will follow Eq. (2.40) and c_u in Eq. (2.42) will be a random number. If this block is set to false, the parametrization of $\boldsymbol{A}$ will follow Eq. (2.42) with c_A specified in the **cA** block. c_u will then be zero.

central charge Sets the charge of a spherical external Coulomb potential. This entry is optional. An external Beryllium-like central potential with Z=4 is included as

```
central charge: 4
```

The charge can also be negative and fractional. Note that this only defines the magnitude of the central charge, but does not introduce the potential-energy term itself. This has to be done later in the **potentials** block.

speed of light Sets the value used for the speed of light to be used in the calculation. This entry is optional and has a default value of 137.0359895 as taken from Ref. [183]. Changing this value to a large value can be used for reproducing the non-relativistic limit in a relativistic calculation:

```
speed of light: 1e5
```

sampling parameters Contains the blocks which define the sampling parameters for the generation of the $\boldsymbol{A}$ matrix and the global vector $\boldsymbol{u}$. This parameter is optional and contains two blocks **a** and **u**. This is an example for the **sampling parameters** block:

```
sampling parameters:
  u:
    mean matrix: " 0.0\n 0.0\n"
    sd matrix: " 1.0\n 1.0\n"
  a:
    sd matrix: " 10.0 5.0\n 5.0 2.5\n"
    mean matrix: " 0.0 0.0\n 0.0 0.0\n"
```

The **a** block contains two matrices. The **mean matrix** contains the mean values and the **sd matrix** contains the standard deviation values for the normal distribution of the sampling of the elements of the $\boldsymbol{A}$ matrix. The matrices are square and the dimension is defined by the number of particle types used in the calculation. Therefore, the above example contains the matrices for a system of two types of particles, e.g, H_2^+.

The **u** block contains two vectors. The **mean matrix** contains the mean values and the **sd matrix** contains the standard deviation values for the normal distribution of the sampling of the elements of the global vector. The dimension of the vectors depends on the number of particle types.

By default, all entries in the two **mean matrix** blocks are set to 1 and in the two **sd matrix** blocks are set to 0.

D.1.2 Further Options

Besides the **composition** block, there are further options which can be set.

size Sets the number of basis functions to be generated. This value is ignored if a parameter set is read from file.

```
size: 100
```

relativistic Is set to true or false. It defines if a non-relativistic or relativistic calculation will be performed. This entry is optional and by default false.

```
relativistic: true
```

polynomials Is either set to true or false. It defines if the polynomial representation of the GVR will be generated. If it set to true, K will be taken as zero.

```
polynomials: true
```

potentials Lists the potentials which will be contained in the Hamiltonian and sets which formula will be used for the evaluation. Possible entries are:

BBCoulomb particle-particle Coulomb interaction evaluated using the formula for ECGs and the GVR (see Eq. (8.60)).

BBCCoulomb particle-particle Coulomb interaction evaluated using the formula for CECGs published by Saito and Suzuki [267]. The **polynomials** entry has to be set to true in order to use this formula.

BBCCoulomb2 particle-particle Coulomb interaction evaluated using the formula for CECGs based on the work of Shiosaki [270] and adapted for explicitly correlated basis functions (see Section 7.2). The **polynomials** entry has to be set to be true in order to use this formula.

BBCCentralPotential central Coulomb potential evaluated using the formula for ECGs and the GVR (see Eq. (8.60)).

BBCCentralPotential central Coulomb potential evaluated using the formula for CECGs published by Saito and Suzuki [267]. The **polynomials** entry has to be set to true in order to use this formula.

BBCCentralPotential2 central Coulomb potential evaluated using the formula for CECGs based on the work of Shiosaki [270] and adapted for explicitly correlated basis functions (see Section 7.2). The **polynomials** entry has to be set to true in order to use this formula.

BBCGaunt Gaunt interaction evaluated using the formula for CECGs published by Saito and Suzuki [267]. The **polynomials** entry has to be set to true in order to use this formula. Always returns zero in non-relativistic calculations.

BBCGaunt2 Gaunt interaction evaluated using the formula for CECGs based on the work of Shiosaki [270] and adapted for explicitly correlated basis functions (see Section 7.2). The **polynomials** entry has to be set to true in order to use this formula. Always returns zero in non-relativistic calculations.

BBCBreit Breit interaction evaluated using the formula for CECGs based on the work of Shiosaki [270] and adapted for explicitly correlated basis functions (see Section 7.2). The **polynomials** entry has to be set to true in order to use this formula. Always returns zero in non-relativistic calculations.

The **potentials** block is used as

```
potentials:
  - BBCCoulomb
  - BBCCentralPotential
```

loewdin Sets the threshold for the selection of eigenvalues in the loewdin ortho-normalization process. This entry is optional and by default $1e{-}12$.

```
loewdin: 1e-13
```

overlap Defines which formula will be used for the evaluation of the overlap integrals. There are two possible values: BBOverlap which uses ECGs and the GVR to evaluate the integrals (see Eq. (8.26)), and BBCOverlap which uses CECGs and uses the formula presented by Saito and Suzuki [267]. BBCOverlap requires that the **polynomials** entry is set to true.

```
overlap: BBOverlap
```

kinetic Defines which formula will be used for the evaluation of the non-relativistic kinetic-energy integrals. There are two possible values: BBKinetic which uses ECGs and the GVR to evaluate the integrals (see Eq. (8.36)), and BBCKinetic for CECGs which uses the formula presented by Saito and Suzuki [267]. BBCKinetic requires that the **polynomials** entry is set to true.

```
kinetic: BBKinetic
```

kmax Sets the maximal value for K in the GVR to be used when basis functions are generated. The values for K will be sampled in a range of 0 to **kmax**. This entry is optional and the default value is 0.

```
kmax: 5
```

v Sets the vibrational quantum number. This entry is optional and the default value is 0.

```
v: 2
```

write to Sets the name of the file in which the basis parameters will be stored at the end of the calculation. This entry is optional.

```
write to: HT+L1Sp1.yml
```

read from Sets the name of the file from which the basis parameters will be read at the beginning of the calculation. This entry is optional.

```
read from: HT+L1Sp1.yml
```

seed Sets the seed value for the generation of the random numbers. This value is optional and a different value is generated by default each time BlueBerry is started. If this value is set to some value, the sampling will always produce the same series of random numbers. This allows

the user to reproduce the results of previous calculations as long they know the value which was used. The value will be stored in the log.yml file and displayed on the screen during the calculation.

```
seed: 5528327
```

write integrals Sets the name of the file which is used to store the integral matrices at the end of a calculation. This entry is optional and by default no integrals are stored.

```
write integrals: HT+L1Sp1_integrals.yml
```

read integrals Sets the name of the file from which integral matrices will be read from at the beginning of a calculation. This entry is optional and by default no integrals are read in but calculated.

```
read integrals: HT+L1Sp1_integrals.yml
```

D.1.3 procedure Block

The procedure block is optional and specifies what calculations and operations will be performed after the initial calculation. The following options are defined for both non-relativistic and relativistic calculations:

sample Optimizes the parameter set by sampling the $\boldsymbol{A}$ matrices.

```
- sample:
    steps: 100
```

steps sets the number of basis functions which will be individually sampled, and not how many times the complete basis set will be sampled. The sampling order is back to front, such that new basis functions will be sampled first.

uniform steps Optimized the parameter set by performing random steps for the parameters of $\boldsymbol{A}$. The step sizes are randomly generated from a uniform distribution. A step will only be performed if the resulting parameter set has a lower energy compared to the original one.

```
- uniform steps:
    steps: 100
    range: 1e-7
```

steps sets the number of basis functions which will be sampled and **range** sets the maximum step size.

save Saves the current intermediate parameter set.

```
- save
```

The parameter set is saved in a file which is named through a time-stamp.

ccr Performs a complex coordinate rotation of the current parameter set. It uses the precalculated integrals and therefore only involves the solution of the complex generalized eigenproblem. The resulting points in the complex spectrum will be written to two files. The first one contains the real values (file extension .real) and the second one contains the imaginary values (file extension .imag).

```
- ccr:
      angle: 0.1
      write to: results
```

Here, the results will be written to the files results.real and results.imag.

debug Starts the debug section of the program. This allows the developer to test parts of the program without major modifications.

```
- debug
```

These options are only defined for non-relativistic calculations:

resize Changes the size of the parameter set. If the new number of basis functions is larger than the original one, additional basis functions will be appended. If the new number of basis functions is lower than the original one, then basis functions will be removed from the back.

```
- resize:
      size: 500
```

cap Includes a complex absorption potential for the study of resonances. The complex absorption potential is a simple r^2 type potential. The resulting points in the complex spectrum will be written to two files. The first one contains the real values (file extension .real) and the second one contains the imaginary values (file extension .imag).

```
- cap:
      angle: 0.4
      write to: results
```

Here, the results will be written to the files results.real and results.imag.

transition dipole Calculates the transition dipole moment to some final state. The parameter set will be read from an external file.

```
- transition dipole:
      read from: final.yml
```

The parameter set for the final state will be read from final.yml.

change spin Changes the spin state of the different particle groups. It is either possible to change the spin state individually for each particle ensemble as

```
- change spin:
      total: false
      particles:
           type: E
           s: 0
           ms: 0
           type: H1
           s:1
           ms: 1
```

or the total spin state for all particles can be generated as

```
- change spin:
      total: true
      s: 0
      ms: 0
```

change angular Changes the angular momentum state of the basis. This allows the user to switch between basis functions which are eigenfunctions of the total spatial angular momentum L:

```
- change angular:
      total: false
      j: 0
      mj: 0
```

or the total angular momentum $J = L + S$

```
- change angular:
      total: true
      j: 2
      mj: 0
      s: 2
      l: 0
```

publication Writes the parameter set to a specific file in a format suitable for publication.

```
- publication:
      write to: publication.yml
```

These options are exclusively defined for relativistic calculations:

analyze state Prints the energy contributions of the $\boldsymbol{\sigma} \cdot \boldsymbol{p}$, mc^2 and potential-energy operators to the total energy of a vibrational state.

```
- analyze state:
      state: 0
```

non-relativistic test Performs a non-relativistic calculation and calculates the relativistic effect from the difference of the non-relativistic energy and the relativistic energy.

```
- non-relativistic test
```

D.1.4 Examples

Here we present some examples for input files.

D.1.4.1 Stochastic Optimization

The first example in Listing D.2 creates and optimizes a parameter set for the hydrogen molecule in its para proton-spin state. We use 500 basis functions and calculate the ground-state energy. 500 sampling steps are performed, i.e., the parameter set is sampled once fully. The final parameter set is saved in a file. We do not use CECGs, but ECGs with the GVR. Since it is a pre-BO calculation, the potential-energy only contains the interactions among all pairs of particles.

Listing D.2: A BlueBerry input file for the hydrogen Molecule in its para proton-spin state. The number of basis functions is 500. The quantum numbers are $L = 0$, $M_L = 0$ and $v = 1$. The parameter set is stored in the file H2Sp1Se0L0v1.yml at the end of the calculation.

```
---
composition:
    particles:
      - type: E
        number: 2
        s: 0
        ms: 0
      - type: H1
        number: 2
        s: 2
        ms: 0
    angular momentum:
      j: 0
      mj: 0
    cA: 1
    born oppenheimer: false
size: 500
v: 1
loewdin: 1.0e-12
potential:
    - BBCoulomb
kinetic: BBKinetic
overlap: BBOverlap
write to: H2Sp1Se0L0v1.yml
procedure:
    - sample:
        steps: 500
```

Listing D.3: A BlueBerry input file for the calculation of transition dipole moment between two rotational states of HT^+ The initial state is read from HT+L0.yml and the final state from HT+L1.yml.

```
---
composition:
    particles:
      - type: E
        number: 1
        s: 1
        ms: 1
      - type: H1
        number: 1
        s: 1
        ms: 1
      - type: H3
        number: 1
        s: 1
        ms: 1
    angular momentum:
      j: 0
      mj: 0
    cA: 1
    born oppenheimer: false
size: 500
loewdin: 1.0e-12
potential:
    - BBCoulomb
kinetic: BBKinetic
overlap: BBOverlap
read from: HT+L0.yml
procedure:
    - transition dipole moment:
        final state: HT+L1.yml
```

D.1.4.2 Transition Dipole Moment

Listing D.3 presents an input file for a calculation of a transition dipole moment for two rotational states of HT^+.

D.1.4.3 Relativistic Calculation

Listing D.4 presents an input file for a relativistic calculation of the hydrogen atom treated as a two-particle system. The number of basis functions is 200. After the initial calculation, the energy contributions of the different terms in the Hamiltonian are listed and a non-relativistic calculation is performed in order to check the non-relativistic limit and calculate the relativistic effect. Note that the **potential** block is stored in the parameter set file. Therefore, if a non-relativistic calculation is used for the generation of the initial parameter set, it is important to use either BBCCoulomb or BBCCoulomb2.

D.1.5 Starting the Calculation

The calculation is started as

```
BlueBerry_No0012 input.yml 10
```

where input.yml is the input file and 10 is the number of cores used in the calculation.

Listing D.4: A BlueBerry input file for a relativistic calculation of the hydrogen atom in the total angular momentum state $J = 1$ $M_J = 1$ generated from the $L = 0$ and $S = 1$ states through LS coupling. 200 basis functions are used in the calculation. It is mandatory that the polynomial representation is created. After the initial calculation, a non-relativistic calculation is performed to find the non-relativistic limit. In the end, the parameter set is written to HJ1_200.yml.

```
---
composition:
    particles:
      - type: E
        number: 1
        s: 1
        ms: 1
      - type: H1
        number: 1
        s: 1
        ms: 1
    angular momentum:
      j: 2
      mj: 0
      l: 0
      s: 2
    cA: 1
    born oppenheimer: false
size: 200
loewdin: 1.0e-12
polynomials: true
potential:
    - BBCCoulomb
write to: HJ1_200.yml
procedure:
    - non-relativistic test
```

Acknowledgments

Foremost, I would like to express my gratitude to *Prof. Dr. Markus Reiher* for giving me the opportunity to work in his research group and to be allowed to work on this interesting subject. His wide knowledge and experience proved to be helpful on many occasions.

Furthermore, I would like to specially thank *Dr. Edit Mátyus*. She provided me with important advice which helped me during the course of this project.

Next I would like to thank all current and former members of the Reiher research group for creating a pleasant working atmosphere. My special thanks goes to my colleagues in charge of maintaining our IT environment and to *Romy Isenegger*, our administrative assistant.

I would also like to thank my family. Their constant support made is possible for me to achieve this degree.

Last, I would like to thank my lovely wife for being patient with me and supporting me during the last three years.

List of Publications

The following publications are included in parts or in an extended version in this thesis:

- B. Simmen, M. Reiher, Relativistic Quantum Theory of Many-Electron Systems.
  In: *M*any-Electron Approaches in Physics, Chemistry and Mathematics; A Multidisciplinary View, edited by V. Bach and L. Delle Site, p. 3, Springer, 2014.
  Chapter: 3

- B. Simmen, E. Mátyus and M. Reiher, "Elimination of the Translational Kinetic Energy Contamination in pre-Born-Oppenheimer Calculations."
  In: *Mol. Phys.* 141 (2014) 154105.
  Chapter 4

- B. Simmen, E. Mátyus and M. Reiher, "Electric Transition Dipole Moment in pre-Born–Oppenheimer Molecular Structure Theory."
  In: *J. Chem. Phys.* 111 (2013) 2086.
  Chapter: 5

- B. Simmen, E. Mátyus and M. Reiher, "Relativistic Kinetic Balance Condition for Explicitly Correlated Basis Functions."
  To be submitted.
  Chapter: 6, B.1, B.2

- B. Simmen, E. Mátyus and M. Reiher, "Pre-Born–Oppenheimer Dirac–Coulomb Fine-Structure Spectrum of Hydrogen-Like Ions."
  To be submitted.
  Chapter: 7, 2

Other publications not part of this work are

- B. Simmen, T. Weymuth and M. Reiher, ”How many Chiral Centers can Raman Optical Activity Spectroscopy Distinguish in a Molecule.” In: *J. Phys. Chem. A* 116 (2012) 5410.

Benjamin Simmen

Date of Birth	27.06.2014
Place of Birth	Baden, Switzerland
Nationality	Swiss and Italian

Education

08.2011 – Present	PhD in Theoretical Chemistry, ETH Zürich, Switzerland Thesis: Developments for a Relativistic Many-1/2-Fermion Theory. Supervisor: Prof. Dr. Markus Reiher
2010 – 2011	Master of Science in Chemistry, ETH Zürich, Switzerland Thesis: How Many Chiral Centers can Raman Optical Activity Spectroscopy Distinguish in a Molecule? Supervisor: Prof. Dr. Markus Reiher
2007 – 2010	Bachelor of Science in Chemistry, ETH Zürich, Switzerland

Experience

08.2011 – Present	Teaching Assistant, ETH Zürich, Switzerland
2008 – 2009	Tutor at Lernpodium Wettingen
2006 – 2007	National Service – Military Service

Talks

03.2014	Developments for a Relativistic Four-Component Many-1/2-Fermion Theory, APS March Meeting, Denver CO, USA
05.2014	Pre-Born-Oppenheimer Theory with Explicitly Correlated Wave Functions, PC-Kolloquien, Zürich, Switzerland

Bibliography

[1] Kong, L., Bischoff, F., Valeev, E. F., Explicitly correlated R12/F12 methods for electronic structure., *Chem. Rev.*, **112**(1) (2012) 75.

[2] Suzuki, Y., Varga, K. *Stochastic Variational Approach to Quantum-Mechanical Few-Body Problems*, Volume 54 of *Lect. Notes Phys.* Springer, Berlin, Heidelberg, Juni 1998.

[3] Bubin, S., Pavanello, M., Tung, W.-C., Sharkey, K. L., Adamowicz, L., Born–Oppenheimer and Non-Born–Oppenheimer, Atomic and Molecular Calculations with Explicitly Correlated Gaussians., *Chem. Rev.*, **113**(1) (2013) 36.

[4] Born, M., Huang, K. *Dynamical theory of crystal lattices*. Clarendon Press, Oxford, UK, 1954.

[5] Born, M., Coupling of electronic and nuclear motion in molecules and crystals, *Nachr. Akad. Wiss. Göttingen Math.-Phys. Kl. II*, **6** (1951) 3.

[6] Born, M., Oppenheimer, R., Zur quantentheorie der molekeln, *Ann. der Phys.*, **84** (1927) 457.

[7] Reiher, M., Wolf, A. *Relativistic Quantum Chemistry*. WILEY-VCH, Weinheim, 2nd ed., 2014.

[8] Dyall, K. G., Fægri, K. *Introduction to Relativistic Quantum Chemistry*. Oxford University Press, 2007.

[9] Simmen, B., Reiher, M. Relativistic Quantum Theory of Many-Electron Systems. In *Many-Electron Approaches in Physics, Chemistry and Mathematics*, Bach, V., Delle Site, L., Eds., p. 3–29. Springer, 2014.

[10] Barysz, M., Ishikawa, Y. *Relativistic Methods for Chemists*, Volume 10 of *Challenges and Advances in Computational Chemistry and Physics*. Springer Science+Business Media, Dordrecht, 2010.

[11] Hirao, K., Ishikawa, Y. *Recent Advances in Relativistic Molecular Theory*. World Scientific, 2004.

[12] Schwerdtfeger, P. *Relativistic Electronic Structure Theory - Fundamentals*. Elsevier Science B.V., 2002.

[13] Hess, B. A. *Relativistic Effects in Heavy-Element Chemistry and Physics*. John Wiley & Sons, Ltd, Chichester, 2003.

[14] Lee, Y. S., McLean, A. D., Relativistic Effects on Re and De in AgH and AuH from All-Electron DiracHartreeFock Calculations, *J. Chem. Phys.*, **76**(1) (1982) 735.

[15] Stanton, R. E., Havriliak, S., Kinetic balance: A partial solution to the problem of variational safety in Dirac calculations, *J. Chem. Phys.*, **81**(4) (1984) 1910.

[16] Dyall, K. G., Grant, I. P., Wilson, S., Matrix Representation of Operator Products, *J. Phys. B*, **493**(17) (1984) 493.

[17] Ishikawa, Y., Binning, R. C., Sando, K. M., Dirac-Fock Discrete-Basis Calculations on the Beryllium Atom, *Chem. Phys. Lett.*, **101**(1) (1983) 111.

[18] Pestka, G., Upper Bounds to the Eigenvalues of the Dirac Hamiltonian, *Phys. Script.*, **203** (2004) 203.

[19] Sun, Q., Liu, W., Kutzelnigg, W., Comparison of restricted, unrestricted, inverse and dual kinetic balances for four-component relativistic calculations, *Theor. Chem. Acc.*, **129** (2011) 423.

[20] Kutzelnigg, W., Completeness of a kinetically balanced Gaussian basis, *J. Chem. Phys.*, **126** (2007) 201103.

[21] Kutzelnigg, W., Basis set expansion of the Dirac operator without variational collapse, *Int. J. Quantum Chem.*, **25** (1984) 107.

[22] Heully, J.-l., Lindgren, I., Lindroth, E., Lundqvist, S., Mårtensson-Pendrill, A.-M., Diagonalisation of the Dirac Hamiltonian as a basis for a relativistic many-body procedure, *J. Phys. B: At. Mol. Phys.*, **19** (1986) 2799.

[23] Shabaev, V., Tupitsyn, I., Yerokhin, V., Plunien, G., Soff, G., Dual Kinetic Balance Approach to Basis-Set Expansions for the Dirac Equation, *Phys. Rev. Lett.*, **93**(13) (2004) 130405.

[24] Kutzelnigg, W., Completeness of a kinetically balanced Gaussian basis, *J. Chem. Phys.*, **126** (2007) 201103.

[25] Dyall, K. G., A question of balance: Kinetic balance for electrons and positrons, *Chem. Phys.*, **395** (2012) 35.

[26] Sutcliffe, B. T. Chapter 31 Coordinate Systems and Transformations. In *Handbook of Molecular Physics and Quantum Chemistry*, Wilson, S., Eds., p. 485. John Wiley & Sons, Ltd, Chichester, 2003.

[27] Kozlowski, P. M., Adamowicz, L., Equivalent quantum approach to nuclei and electrons in molecules, *Chem. Rev.*, **93**(6) (1993) 2007.

[28] Bethe, H. A., Salpeter, E. E. *Quantum Mechanics of One- and Two-Electron Atoms*. Springer, 1977.

[29] Mátyus, E., On the Calculation of Resonances in pre-BornOppenheimer Molecular Structure Theory, *J. Phys. Chem. A*, **117** (2013) 7195.

[30] Mátyus, E., Reiher, M., Molecular Structure Calculations: A Unified Quantum Mechanical Description of Electrons and Nuclei Using Explicitly correlated Gaussian functions and the global vector representation., *J. Chem. Phys.*, **137**(2) (2012) 024104.

[31] Mitroy, J., Bubin, S., Horiuchi, W., Suzuki, Y., Adamowicz, L., Cencek, W., Szalewicz, K., Komasa, J., Blume, D., Varga, K., Theory and application of explicitly correlated Gaussians, *Rev. Mod. Phys.*, **85**(2) (2013) 693.

[32] Henderson, H. V., Searle, S. R., On Deriving the Inverse of a Sum of Matrices, *SIAM Rev.*, **23**(1) (2008) 53.

[33] Mayer, I., Rokob, T., Internal Coordinates of Quantum-Mechanical Systems, *Phys. Rev. A*, **85**(4) (2012) 44101.

[34] Cafiero, M., Bubin, S., Adamowicz, L., Non-Born–Oppenheimer Calculations of Atoms and Molecules, *Phys. Chem. Chem. Phys.*, **5**(8) (2003) 1491.

[35] Suzuki, Y., Usukura, J., Varga, K., New description of orbital motion with arbitrary angular momenta, *J. Phys. B*, **31**(1) (1998) 31.

[36] Varga, K., Suzuki, Y., Usukura, J., Global-Vector Representation of the Angular Motion of Few-Particle Systems, *Few-Body Syst.*, **24**(1) (1998) 81.

[37] Boys, S. F., Electronic Wave Functions. I. A General Method of Calculation for the Stationary States of Any Molecular System, *Proc. Roy. Soc. London A*, **200** (1950) 542.

[38] Kinghorn, D. B., Integrals and derivatives for correlated Gaussian functions using matrix differential calculus, *Int. J. Quantum Chem.*, **57**(2) (1996) 141.

[39] Magnus, J., Neudecker, H., Matrix differential calculus with applications to simple, Hadamard, and Kronecker products, *J. Math. Psych.*, **414492** (1985) 474.

[40] Magnus, J., Neudecker, H. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, 1999.

[41] Varga, K., Suzuki, Y., Stochastic variational method with a correlated Gaussian basis., *Phys. Rev. A*, **53**(3) (1996) 1907.

[42] Varga, K., Suzuki, Y., Precise solution of few-body problems with the stochastic variational method on a correlated Gaussian basis, *Phys. Rev. C*, **52**(6) (1995) 2885.

[43] Mastalerz, R., Reiher, M. Relativistic Electronic Structure Theory for Molecular Spectroscopy. In *Handbook of High Resolution Spectroscopy*, Merkt, F., Quack, M., Eds., p. 405–442, Chichester, 2011. Wiley.

[44] Eliav, E., Kaldor, U. Relativistic four-component multireference coupled cluster methods: Towards a covariant approach. In *Recent Progress in Coupled Cluster Methods*, Carsky et al., P., Eds., p. 113–144. Springer Science+Business, 2010.

[45] Liu, W., Ideas of relativistic quantum chemistry, *Mol. Phys.*, **108**(13) (2010) 1679.

[46] Belpassi, L., Storchi, L., Quiney, H. M., Tarantelli, F., Recent advances and perspectives in four-component Dirac–Kohn–Sham calculations, *Phys. Chem. Chem. Phys.*, **13** (2011) 12368–12394.

[47] Reiher, M., Relativistic Douglas–Kroll–Hess Theory, *WIREs: Comp. Mol. Sci.*, **2** (2012) 139–149.

[48] Peng, D., Reiher, M., Exact decoupling of the relativistic Fock operator, *Theor. Chem. Acc.*, **131**(1) (2012) 1081.

[49] Nakajima, T., Hirao, K., The Douglas–Kroll–Hess Approach, *Chem. Rev.*, **112** (2012) 385–402.

[50] Autschbach, J., Perspective: Relativistic effects, *J. Chem. Phys.*, **136** (2012) 150902.

[51] Wang, D., van Gunsteren, W. F., Chai, Z., Recent advances in computational actinoid chemistry, *Chem. Soc. Rev.*, **41** (2012) 5836–5865.

[52] Pyykkö, P., Relativistic effects in chemistry: more common than you thought., *Ann. Rev. Phys. Chem.*, **63** (2012) 45.

[53] Fleig, T., Relativistic wave-function based electron correlation methods, *Chem. Phys.*, **395** (2012) 2–15.

[54] Schwerdtfeger, P. Relativity and Chemical Bonding. In *The Chemical Bond*, Frenking, G., Shaik, S., Eds., p. 383–404, Weinheim, 2013. Wiley-VCH.

[55] Dirac, P. A. M., The Quantum Theory of the Electron., *Proc. Roy. Soc. London A*, **117** (1928) 610.

[56] Dirac, P. A. M., The Quantum Theory of the Electron (Part II)., *Proc. Roy. Soc. London A1*, **118** (1928) 352.

[57] Esteban, M. J., Lewin, M., Séré, E., Variational methods in relativistic quantum mechanics, *Bull. Amer. Math. Soc.*, **45** (2008) 535.

[58] Pestka, G., Bylicki, M., Karwowski, J., Geminals in Dirac–Coulomb Hamiltonian eigenvalue problem, *J. Math. Chem.*, **50**(3) (2011) 510.

[59] Kutzelnigg, W., Solved and unsolved problems in relativistic quantum chemistry, *Chem. Phys.*, **395** (2012) 16.

[60] Mittleman, M., Configuration-Space Hamiltonian for Heavy Atoms and Correction to the Breit Interaction, *Phys. Rev. A*, **5**(6) (1972) 2395.

[61] Mittleman, M., Theory of relativistic effects on atoms: Configuration-space Hamiltonian, *Phys. Rev. A*, **24**(3) (1981) 1167.

[62] Sucher, J., Foundations of the relativistic theory of many-electron atoms, *Phys. Rev. A*, **22**(2) (1980) 348.

[63] Sucher, J., Erratum: Foundations of the relativistic theory of many-electron atoms, *Phys. Rev. A*, **23**(1) (1981) 388.

[64] Heully, J.-L., Lindgren, I., Lindroth, E., Mårtensson-Pendrill, A.-M., Comment on relativistic wave equations and negative-energy states, *Phys. Rev. A*, **33**(6) (1986) 4426.

[65] Sucher, J., Relativistic many-electron Hamiltonians, *Phys. Scr.*, **36** (1987) 271.

[66] Indelicato, P., Desclaux, J. P., Projection operator in the multiconfiguration Dirac-Fock method, *Phys. Scr.*, **T46** (1993) 110.

[67] Indelicato, P., Projection operators in multiconfiguration Dirac-Fock calculations: Application to the ground state of heliumlike ions, *Phys. Rev. A*, **51**(2) (1995) 1132.

[68] Saue, T., Relativistic Hamiltonians for chemistry: a primer., *Chem. Phys. Chem.*, **12**(17) (2011) 3077.

[69] Liu, W., Perspectives of relativistic quantum chemistry: the negative energy cat smiles., *Phys. Chem. Chem. Phys.*, **14**(1) (2012) 35.

[70] van Wüllen, C., Negative energy states in relativistic quantum chemistry, *Theor. Chem. Acc.*, **131**(1) (2012) 1082.

[71] Pestka, G., Karwowski, J., Dirac-Coulomb Hamiltonian in N-Electron Model Spaces, *Collect. Czech. Chem. Commun.*, **68**(2) (2003) 275.

[72] Pestka, G., Bylicki, M., Karwowski, J., Complex coordinate rotation and relativistic Hylleraas-CI: helium isoelectronic series, *J. Phys. B*, **40**(12) (2007) 2249.

[73] Hylleraas, E. A., Neue Berechnung der Energie des Heliums im Grundzustande, sowie des tiefsten Terms von Ortho-Helium, *Z. Phys.*, **54**(5-6) (1929) 347.

[74] Hylleraas, E. A., Undheim, B., Numerische Berechnung der 2S-Terme von Ortho- und Par-Helium, *Z. Phys.*, **65**(11) (1930) 759.

[75] Green, L., Mulder, M., Milner, P., Correlation Energy in the Ground State of He I, *Phys. Rev.*, **91**(1) (1953) 35.

[76] Chandrasekhar, S., Elbert, D., Herzberg, G., Shift of the 1S1 State of Helium, *Phys. Rev.*, **91**(5) (1953) 1172.

[77] Jastrow, R., Many-Body Problem with Strong Forces, *Phys. Rev.*, **98**(5) (1955) 1479–1484.

[78] Chandrasekhar, S., Herzberg, G., Energies of the Ground States of He, Li^+, and O^{6+}, *Phys. Rev.*, **98**(4) (1955) 1050.

[79] Boys, S. F., The Integral Formulae for the Variational Solution of the Molecular Many-Electron Wave Equations in Terms of Gaussian Functions with Direct Electronic Correlation, *Proc. R. Soc. London Ser. A*, **258**(1294) (1960) 402.

[80] Singer, K., The Use of Gaussian (Exponential Quadratic) Wave Functions in Molecular Problems. I. General Formulae for the Evaluation of Integrals, *Proc. R. Soc. London Ser. A*, **258**(1294) (1960) 412.

[81] Simmen, B., Mátyus, E., Reiher, M., Elimination of the Translational Kinetic Energy Contamination in pre-Born-Oppenheimer Calculations, *Mol. Phys.*, **111**(14-15) (2013) 2086.

[82] Mátyus, E., Hutter, J., Müller-Herold, U., Reiher, M., On the emergence of molecular structure, *Phys. Rev. A*, **83**(5) (2011) 52512.

[83] Mátyus, E., Hutter, J., Müller-Herold, U., Reiher, M., Extracting elements of molecular structure from the all-particle wave function., *J. Chem. Phys.*, **135**(20) (2011) 204302.

[84] Slater, J., Atomic Shielding Constants, *Phys. Rev.*, **36**(1) (1930) 57.

[85] Klopper, W., Manby, F. R., Ten-No, S., Valeev, E. F., R12 methods in explicitly correlated molecular electronic structure theory, *Int. Rev. Phys. Chem.*, **25**(3) (2006) 427.

[86] Ten-no, S., Noga, J., Explicitly correlated electronic structure theory from R12/F12 ansätze, *WIREs Comput. Mol. Sci.*, **2** (2012) 114–125.

[87] Reiher, M., Hinze, J. Four-component Ab Initio Methods for Atoms, Molecules and Solids. In *Relativistic Effects in Heavy-Element Chemistry and Physics*, Hess, B. A., Eds., Chapter 2, p. 61. Wiley, Weinheim, 2003.

[88] Saue, T., Visscher, L. Four-component electronic structure methods for molecules. In *Theoretical chemistry and physics of heavy and superheavy elements*, Wilson, S., Kaldor, U., Eds., Chapter 6, p. 211. Kluwer, Dordrecht, 2003.

[89] Eliav, E., Kaldor, U. Four-component electronic structure methods. In *Relativistic Methods for Chemists*, Barysz, M., Ishikawa, Y., Eds., Chapter 7, p. 279. Springer, Berlin, Heidelberg, 2010.

[90] Schwarz, W. H. E., Wechsel-Trakowski, E., The two problems connected with Dirac-Breit-Roothaan calculations, *Chem. Phys. Lett.*, **85**(1) (1982) 94.

[91] Talman, J. D., Minimax Principle for the Dirac Equation, *Phys. Rev. Lett.*, **57**(9) (1986) 1091–1094.

[92] Kolakowska, A., Talman, J. D., Aashamar, K., Minimax variational approach to the relativistic two-electron problem., *Phys. Rev. A*, **53**(1) (1996) 168.

[93] Fægri Jr., K., Relativistic Gaussian basis sets for the elements K - Uuo, *Theor. Chem. Acc.*, **105**(3) (2001) 252.

[94] Klopper, W., Bak, K. L., Jørgensen, P., Olsen, J., Helgaker, T., Highly accurate calculations of molecular electronic structure, *J. Phys. B*, **32**(13) (1999) R103.

[95] Roothaan, C. C. J., New Developments in Molecular Orbital Theory., *Rev. Mod. Phys.*, **23** (1951) 69.

[96] Hall, G. G., The molecular orbital theory of chemical valency. VIII. A method of calculating ionization potentials., *Proc. Roy. Soc. London A*, **205** (1951) 541.

[97] Møller, C., Plesset, M. S., Note on an Approximation Treatment for Many-Electron Systems., *Phys. Rev.*, **46** (1934) 618.

[98] Binkley, J. S., Pople, J. A., Møller–Plesset Theory for Atomic Ground State Energies., *Int. J. Quantum Chem.*, **9** (1975) 229.

[99] Dyall, K. G., Second-Order Møller–Plesset perturbation theroy for molecular Dirac–Hartree–Fock wavefunctions. Theory for up to two open-shell electrons., *Chem. Phys. Lett.*, **224** (1994) 186.

[100] Laerdahl, J. K., Saue, T., Fægri Jr., K., Direct Relativistic MP2: properties of ground state CuF, AgF and AuF., *Theor. Chem. Acc.*, **97**(1-4) (1997) 177.

[101] Yanai, T., Harrison, R. J., Nakajima, T., Ishikawa, Y., Hirao, K., New Implementation of molecular double point group symmetry in four-component relativistic Gaussian-type spinors., *Int. J. Quantum Chem.*, **107**(6) (2006) 1382.

[102] Buenker, R. J., Peyerimhoff, S. D., CI Method for the study of general molecular potentials., *Theoret. Chim. Acta (Berl.)*, **12** (1968) 183.

[103] Langhoff, S. R., Davidson, E. R., Configuration Interaction calculations on the nitrogen molecule., *Int. J. Quantum Chem.*, **8** (1974) 61.

[104] Buenker, R. J., Peyerimhoff, S. D., Individualized Configuration Selection in CI Calculations with Subsequent Energy Extrapolation., *Theoret. Chim. Acta (Berl.)*, **35** (1974) 33.

[105] Visser, O., Visscher, L., Aerts, P. J. C., Nieuwpoort, W. C., Molecular open shell configuration interaction calculations using the Dirac–Coulomb Hamiltonian: The f^6-manifold of an embedded EuO_6^{9-} cluster, *J. Chem. Phys.*, **96**(4) (1992) 2910–2919.

[106] Fleig, T., Olsen, J., Visscher, L., The generalized active space concept for the relativistic treatment of electron correlation. II. Large-scale configuration interaction implementation based on relativistic 2- and 4-spinors and its application, *J. Chem. Phys.*, **119**(6) (2003) 2963.

[107] Fleig, T., Jensen, H. J. A., Olsen, J., Visscher, L., The generalized active space concept for the relativistic treatment of electron correlation. III. Large-scale configuration interaction and multiconfiguration self-consistent-field four-component methods with application to UO[sub 2], *J. Chem. Phys.*, **124**(10) (2006) 104106.

[108] Knecht, S., Jensen, H. J. A., Fleig, T., Large-scale parallel configuration interaction. II. Two- and four-component double-group general active space implementation with application to BiH., *J. Chem. Phys.*, **132**(1) (2010) 014108.

[109] Hinze, J., MC-SCF. I. The multi-configuration self-consistent-field method., *J. Chem. Phys.*, **59** (1973) 6424.

[110] Werner, H.-J., Matrix-Formulated Direct Multiconfiguration Self-Consistent Field and Multiconfiguration Referece Configuration-Interaction Methods., *Adv. Chem. Phys.*, **69** (1987) 1.

[111] Shepard, R., The Multiconfiguration Self-Consistent Field Method., *Adv. Chem. Phys.*, **69** (1987) 63.

[112] Dyall, K. G., Grant, I. P., Johnson, C. T., Parpia, F. A., Plummer, E. P., GRASP: A general-purpose relativistic atomic structure program, *Comp. Phys. Commun.*, **55** (1989) 425–456.

[113] Jensen, H. J. A., Dyall, K. G., Saue, T., Fægri Jr., K., Relativistic four-component multi-configurational self-consistent-field theory for molecules: Formalism., *J. Chem. Phys.*, **104**(11) (1996) 4083.

[114] Thyssen, J. *Development and Application of Methods for Correlated Relativistic Claculations of Molecular Proerties.* Phd thesis, University of Southern Denmark, Odense, Denmark, 2001.

[115] Eickerling, G., Reiher, M., The shell structure of atoms, *J. Chem. Theory Comput.*, **4** (2008) 286–296.

[116] Thyssen, J., Fleig, T., Jensen, H. J. A., A direct relativistic four-component multiconfiguration self-consistent-field method for molecules., *J. Chem. Phys.*, **129**(3) (2008) 034109.

[117] Roos, B. O., The complete active space self-consistent field method and its applications in electronic structure calculations., *Adv. Chem. Phys.*, **69** (1987) 399.

[118] Andersson, K., Roos, B. O. Multiconfigurational Second-Order Perturbation Theory. In *Modern Electronic Structure Theory Part II*, Yarkony, D., Eds., p. 55. World Scientific, Singapore, 1995.

[119] Abe, M., Gopakmar, G., Nakajima, T., Hirao, K. Relativistic Multireference Perturbation Theory: Complete Active-Space Second-Order Perturbation Theory (CASPT2) with the four-component dirac hamiltonian. In *Chalenges and Advances in Computational Chemistry and Physics*, Leszczynski, J., Eds., p. 157. Springer, 2008.

[120] Cizek, J., On the Correlation Problem in Atomic and Molecular Systems. Calculaion of Wavefunction Components in Ursell-Type Expansion Using Quantum-Field Theoretical Methods., *J. Chem. Phys.*, **45** (1966) 4256.

[121] Lindgren, I., Morrison, J. *Atomic Many-Body Theory*. Springer-Verlag, Berlin, Heidelberg, 2nd ed., 1986.

[122] Lindgren, I., The Coupled-Cluster Approach to Non-Relativistic Many-Body Calculations., *Phys. Scr.*, **36** (1987) 591.

[123] Taylor, P. R. Coupled-Cluster Methods in quantum chemistry. In *Lecture Notes in Quantum Chemistry II*, Roos, B. O., Eds., Chapter 3, p. 125. Springer-Verlag, Berlin, 1994.

[124] Eliav, E., Kaldor, U., Ishikawa, Y., Open-shell relativistic coupled-cluster method with Dirac–Fock–Breit wave functions: Energies of the gold atom and its cation, *Phys. Rev. A*, **49**(3) (1994) 1724–1729.

[125] Visscher, L., Lee, T. J., Dyall, K. G., Formulation and implementation of a relativistic unrestricted coupled-cluster method including noniterative connected triples, *J. Chem. Phys.*, **105**(19) (1996) 8769.

[126] Visscher, L., Eliav, E., Kaldor, U., Formulation and implementation of the relativistic Fock-space coupled cluster method for molecules, *J. Chem. Phys.*, **115**(21) (2001) 9720.

[127] Pernpointner, M., Visscher, L., Parallelization of four-component calculations. II. Symmetry-driven parallelization of the 4-Spinor CCSD algorithm, *J. Comput. Chem.*, **24**(6) (2003) 754.

[128] Beloy, K., Derevianko, A., Application of the dual-kinetic-balance sets in the relativistic many-body problem of atomic structure, *Comput. Phys. Commun.*, **179**(5) (2008) 310.

[129] Mark, F., Schwarz, W. H. E., New Representation of the $\overrightarrow{\alpha} \cdot \overrightarrow{p}$ Operator in the Solution of Dirac-Type Equations by the Linear-Expansion Method, *Phys. Rev. Lett.*, **48**(10) (1982) 673.

[130] Schwarz, W. H. E., Wallmeier, H., Basis set expansions of relativistic molecular wave equations, *Mol. Phys.*, **46**(5) (1982) 1045.

[131] Douglas, M., Kroll, N. M., Quantum Electrodynamical Corrections to the Fine Structure of Helium, *Ann. Phys.*, **82** (1974) 89–155.

[132] Hess, B., Relativistic electronic-structure calculations employing a two-component no-pair formalism with external-field projection operators, *Phys. Rev. A*, **33**(6) (1986) 3742.

[133] Reiher, M., Douglas–Kroll–Hess Theory: a relativistic electrons-only theory for chemistry, *Theor. Chem. Acc.*, **116** (2006) 241–252.

[134] Reiher, M., Wolf, A., Exact decoupling of the Dirac Hamiltonian. I. General theory., *J. Chem. Phys.*, **121**(5) (2004) 2037.

[135] Reiher, M., Wolf, A., Exact decoupling of the Dirac Hamiltonian. II. The generalized Douglas–Kroll–Hess transformation up to arbitrary order, *J. Chem. Phys.*, **121** (2004) 10945.

[136] Wolf, A., Reiher, M., Hess, B. A., Correlated ab initio calculations of spectroscopic parameters of SnO within the framework of the higher-order generalized Douglas–Kroll transformation, *J. Chem. Phys.*, **120** (2004) 8624–8631.

[137] Wolf, A., Reiher, M., Hess, B. A., The generalized Douglas–Kroll transformation, *J. Chem. Phys.*, **117**(20) (2002) 9215–9226.

[138] Wolf, A., Reiher, M., Exact decoupling of the Dirac Hamiltonian. III. Molecular properties., *J. Chem. Phys.*, **124** (2006) 64102.

[139] Wolf, A., Reiher, M., Exact decoupling of the Dirac Hamiltonian. IV. Automated evaluation of molecular properties within the Douglas–Kroll–Hess theory up to arbitrary order., *J. Chem. Phys.*, **124** (2006) 064102.

[140] Mastalerz, R., Lindh, R., Reiher, M., The Douglas–Kroll–Hess Electron Density at an Atomic Nucleus, *Chem. Phys. Lett.*, **465** (2008) 157–164.

[141] Knecht, S., Fux, S., van Meer, R., Visscher, L., Reiher, M., Saue, T., Mössbauer spectroscopy for heavy elements: a relativistic benchmark study of mercury, *Theor. Chem. Acc.*, **129** (2011) 631–650.

[142] Peng, D., Hirao, K., An arbitrary order Douglas–Kroll method with polynomial cost, *J. Chem. Phys.*, **130** (2009) 44102.

[143] Autschbach, J., Peng, D., Reiher, M., Two-Component Relativistic Calculations of Electric-Field Gradients Using Exact Decoupling Methods: Spinorbit and Picture-Change Effects, *J. Chem. Theory Comput.*, **8** (2012) 4239–4248.

[144] Peng, D., Middendorf, N., Weigend, F., Reiher, M., An efficient implementation of two-component relativistic exact-decoupling methods for large molecules, *J. Chem. Phys.*, **138** (2013) 184105.

[145] Siedentop, H., Stockmeyer, E., An analytic Douglas–Kroll–Heß method, *Phys. Lett. A*, **341**(5-6) (2005) 473.

[146] Siedentop, H., Stockmeyer, E., The Douglas–Kroll–Heß Method: Convergence and Block-Diagonalization of Dirac Operators, *Ann. Henri Poincaré*, **7**(1) (2006) 45.

[147] Barysz, M., Sadlej, A. J., Snijders, J. G., Nonsingular two/one-component relativistic Hamiltonians accurate through arbitrary high order in $\alpha 2$, *Int. J. Quantum Chem.*, **65**(3) (1997) 225.

[148] Barysz, M., Sadlej, A. J., Infinite-order two-component theory for relativistic quantum chemistry, *J. Chem. Phys.*, **116**(7) (2002) 2696.

[149] Barysz, M., Sadlej, A. J., Two-component methods of relativistic quantum chemistry: from the Douglas–Kroll approximation to the exact two-component formalism, *J. Mol. Struct.: THEOCHEM*, **573**(1-3) (2001) 181.

[150] Keçdziera, D., Barysz, M., Non-iterative approach to the infinite-order two-component (IOTC) relativistic theory and the non-symmetric algebraic Riccati equation, *Chemical Physics Letters*, **446**(1-3) (2007) 176.

[151] Foldy, L. L., Wouthuysen, S. A., On the Dirac Theory of Spin 1/2 Particles and its Non-Relativistic Limit, *Phys. Rev.*, **78** (1950) 29.

[152] Baerends, E. J., Schwarz, W. H. E., Schwerdtfeger, P., Snijders, J. G., Relativistic atomic orbital contractions and expansions: magnitudes and explanations, *J. Phys. B*, **23**(19) (1990) 3225.

[153] Filatov, M., Cremer, D., Representation of the exact relativistic electronic Hamiltonian within the regular approximation, *J. Chem. Phys.*, **119**(22) (2003) 11526.

[154] Kutzelnigg, W., Liu, W., Quasirelativistic theory equivalent to fully relativistic theory., *J. Chem. Phys.*, **123**(24) (2005) 241102.

[155] Filatov, M., Cremer, D., Connection between the regular approximation and the normalized elimination of the small component in relativistic quantum theory., *J. Chem. Phys.*, **122**(6) (2005) 064104.

[156] Filatov, M., Dyall, K. G., On convergence of the normalized elimination of the small component (NESC) method, *Theor. Chem. Acc.*, **117**(3) (2006) 333.

[157] Liu, W., Peng, D., Infinite-order quasirelativistic density functional method based on the exact matrix quasirelativistic theory, *J. Chem. Phys.*, **125**(4) (2006) 044102.

[158] Kutzelnigg, W., Liu, W., Quasirelativistic theory I. Theory in terms of a quasi-relativistic operator, *Mol. Phys.*, **104**(13-14) (2006) 2225.

[159] Liu, W., Peng, D., Exact two-component Hamiltonians revisited., *J. Chem. Phys.*, **131**(3) (2009) 031104.

[160] Peng, D., Liu, W., Xiao, Y., Cheng, L., Making four- and two-component relativistic density functional methods fully equivalent based on the idea of "from atoms to molecule"., *J. Chem. Phys.*, **127**(10) (2007) 104106.

[161] Ilias, M., Saue, T., An infinite-order two-component relativistic Hamiltonian by a simple one-step transformation., *J. Chem. Phys.*, **126**(6) (2007) 064102.

[162] Liu, W., Kutzelnigg, W., Quasirelativistic theory. II. Theory at matrix level., *J. Chem. Phys.*, **126**(11) (2007) 114107.

[163] Sikkema, J., Visscher, L., Saue, T., Ilias, M., The molecular mean-field approach for correlated relativistic calculations., *J. Chem. Phys.*, **131**(12) (2009) 124116.

[164] Peng, D., Reiher, M., Local relativistic exact decoupling., *J. Chem. Phys.*, **136**(24) (2012) 244108.

[165] Chang, C., Pelissier, M., Durand, P., Regular Two-Component Pauli-Like Effective Hamiltonians in Dirac Theory, *Phys. Scr.*, **34**(5) (1986) 394.

[166] van Lenthe, E., Baerends, E. J., Snijders, J. G., Relativistic total energy using regular approximations., *J. Chem. Phys.*, **101** (1994) 9783.

[167] van Lenthe, E., Snijders, J. G., Baerends, E. J., The zero-order regular approximation for relativistic effects: The effect of spin-orbit coupling in closed shell molecules., *J. Chem. Phys.*, **105** (1996) 6505.

[168] van Lenthe, E., Baerends, E. J., Snijders, J. G., van Lenthe, E., Relativistic regular two-component Hamiltonians, *J. Chem. Phys.*, **99**(6) (1993) 4597.

[169] van Leeuwen, R., van Lenthe, E., Baerends, E. J., Snijders, J. G., Exact solutions of regular approximate relativistic wave equations for hydrogen-like atoms, *J. Chem. Phys.*, **101**(2) (1994) 1272.

[170] van Lenthe, E., van Leeuwen, R., Baerends, E. J., Snijders, J. G., Relativistic regular two-component Hamiltonians, *Int. J. Quantum Chem.*, **57**(3) (1996) 281.

[171] Sadlej, A. J., Snijders, J. G., van Lenthe, E., Baerends, E. J., Four component regular relativistic Hamiltonians and the perturbational treatment of Diracs equation, *J. Chem. Phys.*, **102**(4) (1995) 1758.

[172] Moncho, S., Autschbach, J., Relativistic Zeroth-Order Regular Approximation Combined with Nonhybrid and Hybrid Density Functional Theory: Performance for NMR Indirect Nuclear SpinSpin Coupling in Heavy Metal Compounds, *J. Chem. Theory Comput.*, **6**(1) (2010) 223–234.

[173] Autschbach, J., The accuracy of hyperfine integrals in relativistic NMR computations based on the zeroth-order regular approximation, *Theor. Chem. Acc.*, **112**(1) (2004) 52.

[174] Brown, G. E., Ravenhall, D. G., On the interaction fo two electrons., *Proc. Roy. Soc. London A*, **208** (1951) 94.

[175] Aguilar, J., Combes, J. M., A Class of Analytic Perturbations for One-Body Schrödinger Hamiltonians, *Comm. Math. Phys.*, **22**(22) (1971) 269.

[176] Balslev, E., Combes, J. M., Spectral properties of many-body Schrödinger operators with dilatation-analytic interactions, *Comm. Math. Phys.*, **22**(4) (1971) 280.

[177] Simon, B., Resonances in n-Body Quantum Systems with Dilatation Analytic Potentials and the Foundations of Time-Dependent Perturbation Theory, *Ann. Math.*, **97** (1973) 247.

[178] Doolen, G. D., A procedure for calculating resonance eigenvalues, *J. Phys. B*, **8**(4) (1975) 525.

[179] Reinhardt, W. P., Complex Coordinates in the Theory of Atomic and Molecular Structure and Dynamics, *Ann. Rev. Phys. Chem.*, **33** (1982) 223.

[180] Ho, Y. K., The Method of Complex Coordinate Rotation and its Applications to Atomic Collision Processes, *Phys. Rep.*, **99**(1) (1983) 1.

[181] Kukulin, V. I., Krasnopol'sky, V. M., Horáček,. *Theory of Resonances*. Kluwer Academic Publishers, Dodrecht, 1989.

[182] Pestka, G., Bylicki, M., Karwowski, J., Application of the complex-coordinate rotation to the relativistic Hylleraas-CI method: a case study, *J. Phys. B*, **39**(14) (2006) 2979.

[183] Pestka, G., Bylicki, M., Karwowski, J. Dirac-Coulomb Equation: Playing with Artifacts. In *Frontiers in Quantum Systems in Chemistry and Physics*, Wilson, S., Grout, P. J., Maruani, J., Delgado-Barrio, G., Piecuch, P., Eds., p. 215. Springer, 2008.

[184] Pyykkö, P., Desclaux, J. P., Relativity and the periodic system of elements, *Acc. Chem. Res.*, **12**(8) (1979) 276.

[185] Pyykkö, P., Relativistic effects in structural chemistry, *Chem. Rev.*, **88**(3) (1988) 563.

[186] Darwin, C. G., On the Diffraction of the Magnetic Electron, *Proc. R. Soc. London Ser. A*, **120**(786) (1928) 631.

[187] Hess, B. A., Perspective on "Zur Quantentheorie der Spektrallinien", *Theor. Chem. Acc.*, **103**(3-4) (2000) 168.

[188] Sommerfeld, A., Zur Quantentheorie der Spektrallinien, *Ann. Phys.*, **356**(17) (1916) 1.

[189] Pestka, G., Tatewaki, H., Karwowski, J., Relativistic correlation energies of heliumlike atoms, *Phys. Rev. A*, **70**(2) (2004) 6.

[190] Parpia, F. A., Grant, I. P., Accurate Dirac-Coulomb Energies for the Ground States of Helium-Like Atoms, *J. Phys. B*, **23** (1990) 211.

[191] Pachucki, K., Yerokhin, V., Reexamination of the helium fine structure, *Phys. Rev. A*, **79**(6) (2009) 062516–1.

[192] Pestka, G., Variational solution of the Dirac-Coulomb equation using explicitly correlated wavefunctions. Matrix elements and radial integrals, *J. Phys. A*, **31**(29) (1998) 6243.

[193] Bylicki, M., Pestka, G., Karwowski, J., Relativistic Hylleraas configuration-interaction method projected into positive-energy space, *Phys. Rev. A*, **77**(4) (2008) 44501.

[194] Ottschofski, E., Kutzelnigg, W., Direct Perturbation Theory of Relativistic Effects for Explicitly Correlated Wave Functions: The He Isoelectronic Series, *J. Chem. Phys.*, **106**(16) (1997) 6634.

[195] Liu, J., Salumbides, E. J., Hollenstein, U., Koelemeij, J. C. J., Eikema, K. S. E., Ubachs, W., Merkt, F., Determination of the ionization and dissociation energies of the hydrogen molecule., *J. Chem. Phys.*, **130**(17) (2009) 174306.

[196] Liu, J., Sprecher, D., Jungen, C., Ubachs, W., Merkt, F., Determination of the ionization and dissociation energies of the deuterium molecule (D(2))., *J. Chem. Phys.*, **132**(15) (2010) 154301.

[197] Sprecher, D., Liu, J., Jungen, C., Ubachs, W., Merkt, F., Communication: The ionization and dissociation energies of HD., *J. Chem. Phys.*, **133**(11) (2010) 111102.

[198] Piszczatowski, K., Łach, G., Przybytek, M., Komasa, J., Pachucki, K., Jeziorski, B., Theoretical Determination of the Dissociation Energy of Molecular Hydrogen, *J. Chem. Theory Comput.*, **5**(11) (2009) 3039.

[199] Pachucki, K., Komasa, J., Rovibrational levels of HD., *Phys. Chem. Chem. Phys.*, **12**(32) (2010) 9188.

[200] Chakraborty, A., Pak, M. V., Hammes-Schiffer, S., Inclusion of Explicit Electron-Proton Correlation in the Nuclear-Electronic Orbital Approach Using Gaussian-Type Geminal Functions., *J. Chem. Phys.*, **129**(1) (2008) 014101.

[201] Bochevarov, A. D., Valeev, E. F., Sherrill, C. D., The Electron and Nuclear Orbitals Model: Current Challenges and Future Prospects, *Mol. Phys.*, **102**(1) (2004) 111.

[202] Goli, M., Shahbazian, S., The Two-Component Quantum Theory of Atoms in Molecules (TC-QTAIM): Foundations, *Theor. Chem. Acc.*, **131**(5) (2012) 1208.

[203] Albert, S., Hollenstein, H., Quack, M., Willeke, M., Rovibrational analysis of the ν_4, $2\nu_6$ Fermi resonance band of $CH^{35}ClF_2$ by means of a polyad Hamiltonian involving the vibrational levels ν_4, $2\nu_6$, $\nu_6+\nu_9$ and $2\nu_9$, and comparison with ab initio calculations., *Mol. Phys.*, **104**(16-17) (2006) 2719.

[204] Albert, S., Albert, K. K., Hollenstein, H., Tanner, C. M., Quack, M. Fundamentals of RotationVibration Spectra. In *High-Resolution Spectroscopy*, Quack, M., Merkt, F., Eds. John Wiley and Sons, Ltd., Chichester, 2011.

[205] Luckhaus, D., Quack, M., Willeke, M., Coupling Across Bonds : Ab Initio Calculations for the Anharmonic Vibrational Resonance Dynamics of the Coupled OH and CH Chromophores in Trans Formic Acid HCOOH, *Z. Physik. Chem.*, **1114**(8) (2000) 1087.

[206] Kuhn, B., Rizzo, T. R., Luckhaus, D., Quack, M., Suhm, M. A., A New Six-Dimensional Analytical Potential up to Chemically Significant Energies for the Electronic Ground State of Hydrogen Peroxide, *J. Chem. Phys.*, **111**(6) (1999) 2565.

[207] Quack, M., Molecules in Motion, *Chimia*, **55**(10) (2001) 753.

[208] Mátyus, E., Czakó, G., Császár, A. G., Toward Black-Box-Type Full- and Reduced-Dimensional Variational (Ro)Vibrational Computations., *J. Chem. Phys.*, **130**(13) (2009) 134112.

[209] Nakai, H., Simultaneous Determination of Nuclear and Electronic Wave Functions Without Born-Oppenheimer Approximation: Ab Initio NO+MO/HF Theory, *Int. J. Quantum Chem.*, **86**(6) (2002) 511.

[210] Nakai, H., Hoshino, M., Miyamoto, K., Hyodo, S., Elimination of Translational and Rotational Motions in Nuclear Orbital Plus Molecular Orbital Theory., *J. Chem. Phys.*, **122**(16) (2005) 164101.

[211] Sutcliffe, B., Comment on "Elimination of Translational and Rotational Motions in Nuclear Orbital Plus Molecular Orbital Theory" [J. Chem. Phys. 122, 164101 (2005)]., *J. Chem. Phys.*, **123**(23) (2005) 237101; author reply 237102.

[212] Jeziorski, B., Szalewicz, K., High-Accuracy Compton Profile of Molecular Hydrogen from Explicitly Correlated Gaussian Wave Function, *Phys. Rev. A*, **19**(6) (1979) 2360.

[213] Cencek, W., Rychlewski, J., Many-Electron Explicitly Correlated Gaussian Functions. I. General theory and Test Results, *J. Chem. Phys.*, **98**(2) (1993) 1252.

[214] Rychlewski, J., Eds. *Explicitly Correlated Wave Functions in Chemistry and Physics*. Kluwer Academic Publishers, 2003.

[215] Tian, Q.-L., Tang, L.-Y., Zhong, Z.-X., Yan, Z.-C., Shi, T.-Y., Oscillator Strengths Between Low-Lying Ro-Vibrational States of Hydrogen Molecular Ions., *J. Chem. Phys.*, **137**(2) (2012) 024311.

[216] Bekbaev, A. K., Korobov, V. I., Dineykhan, M., Variational Calculations of the HT^+ Rovibrational Energies, *Phys. Rev. A*, **83**(4) (2011) 044501.

[217] Simmen, B., Mátyus, E., Reiher, M., Calculation of the Electric Transition Dipole Moment in pre-Born-Oppenheimer Molecular Structure Theory, *J. Chem. Phys.*, **141** (2014) 154105.

[218] Rodríguez, C. G., Urbina, A. S., Torres, F. J., Cazar, D., Ludeña, E. V., Non-Born–Oppenheimer nuclear and electronic densities for a three-particle Hooke–Coulomb model, *Comp. Theor. Chem.*, **1018** (2013) 26.

[219] Müller-Herold, U., On the emergence of molecular structure from atomic shape in the 1/r2 harmonium model., *J. Chem. Phys.*, **124**(1) (2006) 14105.

[220] Müller-Herold, U., On the transition between directed bonding and helium-like angular correlation in a modified Hooke-Calogero model, *Eur. Phys. J. D*, **49**(3) (2008) 311–315.

[221] Ludeña, E. V., Echevarría, L., Lopez, X., Ugalde, J. M., Non-Born–Oppenheimer electronic and nuclear densities for a Hooke-Calogero three-particle model: non-uniqueness of density-derived molecular structure., *J. Chem. Phys.*, **136**(8) (2012) 084103.

[222] Becerra, M., Posligua, V., Ludeña, E. V., Non-Born–Oppenheimer nuclear and electronic densities for a Hooke-Coulomb model for a four-particle system, *Int. J. Quantum Chem.*, **113**(10) (2013) 1584–1590.

[223] Goli, M., Shahbazian, S., The two-component quantum theory of atoms in molecules (TC-QTAIM): the unified theory of localization/delocalization of electrons, nuclei, and exotic elementary particles, *Theor. Chem. Acc.*, **132**(12) (2013) 1410.

[224] Aguirre, N. F., Villarreal, P., Delgado-Barrio, G., Posada, E., Reyes, A., Biczysko, M., Mitrushchenkov, A. O., de Lara-Castells, M. P., Including nuclear quantum effects into highly correlated electronic structure calculations of weakly bound systems., *J. Chem. Phys.*, **138**(18) (2013) 184113.

[225] King, A. W., Longford, F., Cox, H., The stability of S-states of unit-charge Coulomb three-body systems: From H(-) to H2 (+)., *J. Chem. Phys.*, **139**(22) (2013) 224306.

[226] Pérez-Torres, J. F., Electronic flux densities in vibrating H_2^+ in terms of vibronic eigenstates, *Phys. Rev. A*, **87**(6) (2013) 062512.

[227] Sutcliffe, B. T., Woolley, R. G., Comment on Molecular structure in non-BornOppenheimer quantum mechanics, *Chem. Phys. Lett.*, **408**(4-6) (2005) 445.

[228] Sutcliffe, B. T., Woolley, R. G., Molecular structure calculations without clamping the nuclei, *Phys. Chem. Chem. Phys.*, **7**(21) (2005) 3664.

[229] Woolley, R. G., Quantum theory and molecular structure, *Adv. Phys.*, **25**(1) (1976) 27.

[230] Woolley, R. G., Is there a quantum definition of a molecule?, *J. Math. Chem.*, **23** (1998) 3.

[231] Woolley, R. G., Quantum chemistry beyond the Born-Oppenheimer approximation, *Comp. Theor. Chem.*, **230** (1991) 17.

[232] Hilborn, R. C., Einstein Coefficients, Cross Sections, f Values, Dipole Moments, and All That, *Am. J. Phys.*, **50**(11) (1982) 982.

[233] Chandrasekhar, S., On the Continuous Absorption Coefficient of the Negative Hydrogen Ion, *Astrophys. J.*, **102** (1945) 223.

[234] Anderson, M. T., Weinhold, F., Relative Accuracy of Length and Velocity Forms in Oscillator-Strength Calculations, *Phys. Rev. A*, **10**(5) (1974) 1457.

[235] Wolniewicz, L., Staszewska, G., Transition Moments for the Hydrogen Molecule, *J. Mol. Spectrosc.*, **217**(2) (2003) 181.

[236] Mohr, P. J., Taylor, B. N., Newell, D. B., CODATA recommended values of the fundamental physical constants: 2006, *Rev. Mod. Phys.*, **80**(2) (2008) 633.

[237] Pachucki, K., Komasa, J., Nonadiabatic Corrections to Rovibrational Levels of H_2., *J. Chem. Phys.*, **130**(16) (2009) 164113.

[238] Wolniewicz, L., Orlikowski, T., Staszewska, G., $^1\Sigma_u$ and $^1\Pi_u$ States of the Hydrogen Molecule: Nonadiabatic Couplings and Vibrational Levels, *J. Mol. Spectrosc.*, **238**(1) (2006) 118.

[239] Ten-no, S., Yamaki, D., Communication: Explicitly correlated four-component relativistic second-order Møller-Plesset perturbation theory., *J. Chem. Phys.*, **137**(13) (2012) 131101.

[240] Li, Z., Shao, S., Liu, W., Relativistic explicit correlation: coalescence conditions and practical suggestions., *J. Chem. Phys*, **136**(14) (2012) 144117.

[241] Granlund, T. GNU MP: The GNU Multiple Precision Arithmetic Library, 2012. gmplib.org.

[242] Nakata, M. The MPACK (MBLAS/MLAPACK); a multiple precision arithmetic version of BLAS and LAPACK. Technical report. `http://mplapack.sourceforge.net/`.

[243] Fousse, L., Hanrot, G., Lefèvre, V., Pélissier, P., Zimmermann, P., MPFR: A Multiple-Precision Binary Floating-Point Library with Correct Rounding, *ACM T. Math. Software*, **33**(2) (2007) 13.

[244] Francis, J. G. F., The QR Transformation, I, *Comput. J.*, **4**(3) (1961) 265.

[245] Francis, J. G. F., The QR Transformation, II, *Comput. J.*, **4**(4) (1962) 332.

[246] Kublanovskaya, V. N., On some algorithms for the solution of the complete eigenvalue problem, *USSR Comput. Math. & Math. Phys.*, **1**(3) (1963) 637.

[247] Gu, M., Eisenstat, S. C., A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem, *SIAM J. Matrix Anal. Apply.*, **15** (1994) 1266.

[248] Saue, B. T., Fægri, K., Helgaker, T., Gropen, O., Principles of direct 4-component relativistic SCF: application to caesium auride, *Mol. Phys.*, **91**(5) (1997) 937.

[249] Kinghorn, D. B., Adamowicz, L., A correlated basis set for nonadiabatic energy calculations on diatomic molecules, *J. Chem. Phys.*, **110**(15) (1999) 7166.

[250] Bubin, S., Adamowicz, L., Molski, M., An accurate non-Born-Oppenheimer calculation of the first purely vibrational transition in LiH molecule., *J. Chem. Phys.*, **123**(13) (2005) 134310.

[251] Sharkey, K. L., Bubin, S., Adamowicz, L., Lower Rydberg ^{2}D states of the lithium atom: Finite-nuclear-mass calculations with explicitly correlated Gaussian functions, *Phys. Rev. A*, **83**(1) (2011) 12506.

[252] Bubin, S., Adamowicz, L., Variational Calculations of Excited States with Zero Total Angular Momentum (Vibrational Spectrum) of H_2 Without Use of the Born–Oppenheimer Spproximation, *J. Phys. Chem.*, **118**(7) (2003) 3079.

[253] Bubin, S., Leonarski, F., Stanke, M., Adamowicz, L., Non-Adiabatic Corrections to the Energies of the Pure Vibrational States of H_2, *Chem. Phys. Lett.*, **477**(1-3) (2009) 12.

[254] Kozlowski, P. M., Adamowicz, L., An effective method for generating nonadiabatic many-body wave function using explicitly correlated Gaussian-type functions, *J. Chem. Phys.*, **95**(9) (1991) 6681.

[255] Kozlowski, P., Adamowicz, L., Nonadiabatic variational calculations for the ground state of the positronium molecule, *Phys. Rev. A*, **48**(3) (1993) 1903.

[256] Bubin, S., Stanke, M., Kędziera, D., Adamowicz, L., Relativistic corrections to the ground-state energy of the positronium molecule, *Phys. Rev. A*, **75**(6) (2007) 62504.

[257] Kirnosov, N., Sharkey, K. L., Adamowicz, L., Lifetimes of rovibrational levels of HD^2, *Phys. Rev. A*, **89**(1) (2014) 012513.

[258] Freund, D. E., Huxtable, B. D., Morgan III, J. D., Variational Calculations on the Helium Isoelectronic Sequence, *Phys. Rev. A*, **29**(2) (1984) 980.

[259] Dirac, P., The Quantum Mechanics of Many-Electron Systems, *Proc. R. Soc. London Ser. A*, **132** (1929) 714.

[260] Simmen, B., Mátyus, E., Reiher, M., Relativistic Kinetic Balance Condition for Explicitly Correlated Basis Functions, *J. Chem. Phys.*, (2014) To be submitted.

[261] Marsch, E., An effective Dirac equation for a binary of two fermions, *J. Phys. A: Math. Theor.*, **41**(18) (2008) 185301.

[262] Marsch, E., Addendum and erratum, The relativistic energy spectrum of hydrogen, *Ann. der Phys.*, **15**(6) (2006) 434.

[263] Marsch, E., The relativistic energy spectrum of hydrogen, *Ann. der Phys.*, **14**(5) (2005) 324.

[264] Marsch, E., The radial wavefunction of a relativistic binary of two fermions bound by the Coulomb force, *Ann. der Phys.*, **16**(7-8) (2007) 553.

[265] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematic, Philadelphia, PA, third ed., 1999.

[266] Sanderson, C. Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Technical report, NICTA, 2010. `http://arma.sourceforge.net/armadillo_nicta_2010.pdf`.

[267] Saito, S. L., Suzuki, Y.-I., Computational method of many-electron integrals over explicitly correlated Cartesian Gaussian functions, *J. Chem. Phys.*, **114**(3) (2001) 1115.

[268] Obara, S., Saika, A., General recurrence formulas for molecular integrals over Cartesian Gaussian functions, *J. Chem. Phys.*, **89**(3) (1988) 1540.

[269] Obara, S., Saika, A., Efficient recursive computation of molecular integrals over Cartesian Gaussian functions, *J. Chem. Phys.*, **84**(7) (1986) 3963.

[270] Shiozaki, T., Communication: An efficient algorithm for evaluating the Breit and spin-spin coupling integrals., *J. Chem. Phys.*, **138**(11) (2013) 111101.

[271] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, New York, 3 ed., 2007.

[272] Sherman, J., Morrison, W. J., Adjustment of an Inverse Matrix Corresponding to Changes in the Elements of a Given Column or a Given Row of the Original Matrix (abstract), *Ann. Math. Stat.*, **20**(4) (1949) 620–624.

[273] Sherman, J., Morrison, W. J., Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix, *Ann. Math. Stat.*, **21**(1) (1950) 124–127.

[274] Harville, D. A. *Matrix Algebra From a Statistician's Perspective*. Springer, New York, 1997.

[275] Marsch, E. Private Communication, 2014.

[276] Audi, G., Bersillon, O., Blachot, J., Wapstra, A., The NUBASE evaluation of nuclear and decay properties, *Nucl Phys. A*, **729** (2003) 3.

[277] Gordon, M. S., Pople, J. A., Approximate Self-Consistent Molecular-Orbital Theory. VI. INDO Calculated Equilibrium Geometries, *J. Chem. Phys.*, **49** (1968) 4643.

[278] Tracy, S., Singh, P., Partitioned matrix differentiation, *Stat. Neerl.*, **26**(4) (1972) 143.

[279] Wei, Y., Zhang, F., Equivalence of a Matrix Product to the Kronecker Product, *Hadronic J. Suppl.,*, **15**(3) (2000) 327.

[280] Horn, R. A., Mathias, R., Block-matrix generalizations of Schur's basic theorems on Hadamard products, *Linear Algebra Appl.*, **172** (1992) 337.

[281] Wansbeek, T., Block Kronecker Products and the vecb Operator, *Linear Algebra Appl.*, **149** (1991) 165.

[282] Ben-Kiki, O., Evans, C., döt Net, I. YAML Ain't Markup Language, 2009. http://yaml.org/.

[283] Murray-Rust, P., Rzepa, H., Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles, *J. Chem. Inf. Comput. Sci.*, **39**(6) (1999) 928.

[284] Murray-Rust, P., Rzepa, H. S., Chemical markup, XML and the World-Wide Web. 2. Information objects and the CMLDOM., *J. Chem. Inf. Comput. Sci.*, **41**(5) (2001) 1113.

[285] Gkoutos, G. V., Murray-Rust, P., Rzepa, H. S., Wright, M., Chemical markup, XML and the World-Wide Web. 3. Toward a signed semantic chemical web of trust., *J. Chem. Inf. Comput. Sci.*, **41**(5) (2001) 1124.

[286] Murray-Rust, P., Rzepa, H. S., Chemical markup, XML, and the World Wide Web. 4. CML schema., *J. Chem. Inf. Comput. Sci.*, **43**(3) (2003) 757.

[287] Murray-Rust, P., Rzepa, H. S., Williamson, M. J., Willighagen, E. L., Chemical markup, XML, and the World Wide Web. 5. Applications of chemical metadata in RSS aggregators., *J. Chem. Inf. Comput. Sci.*, **44**(2) (2004) 462.

[288] Holliday, G. L., Murray-Rust, P., Rzepa, H. S., Chemical markup, XML, and the world wide web. 6. CMLReact, an XML vocabulary for chemical reactions., *J. Chem. Inf. Model.*, **46**(1) (2006) 145.

[289] Kuhn, S., Helmus, T., Lancashire, R. J., Murray-Rust, P., Rzepa, H. S., Steinbeck, C., Willighagen, E. L., Chemical Markup, XML, and the World Wide Web. 7. CMLSpect, an XML vocabulary for spectral data., *J. Chem. Inf. Model.*, **47**(6) (2015) 2015.

[290] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., Yeregeau, F. Extensible Markup Language (XML) 1.0 (Fifth Edition), 2008. http://www.w3.org/TR/2008/REC-xml-20081126/.

[291] Zhang, X., Wang, Q., Zhang, Y. Model-driven Level 3 BLAS Performance Optimization on Loongson 3A Processor. In *ICPADS*, 2012.

[292] Wang, Q., Zhang, X., Zhang, Y., Yi, Q. AUGEM: automatically generate high performance dense linear algebra kernels on x86 CPUs. In *SC'13*, 2013.

[293] OpenBlas. xianyi.github.com/OpenBLAS.

[294] Intel Math Kernel Library. http://developer.intel.com/software/products/mkl/.

[295] yaml-cpp. https://code.google.com/p/yaml-cpp/.

[296] Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Booth, M., Rossi, F. GNU Scientific Library Reference Manual. Technical report, 2013. `http://www.gnu.org/software/gsl/`.

[297] Josefsson, S. The Base16, Base32, and Base64 Data Encodings. RFC 4648, 2006.

[298] libb64. http://libb64.sourceforge.net/.