
Kramers-Restricted Self-Consistent 2-Spinor Fields for Heavy-Element Chemistry

Inaugural-Dissertation
zur Erlangung des Doktorgrads
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von

Jonas Wiebke
aus Frankfurt a. M.

Köln 2010

Berichterstatter:

Prof. Dr. M. Dolg
Prof. Dr. U. Deiters

Tag der mündlichen Prüfung: 7. Juli 2010

6.52: “*Wir fühlen, dass selbst, wenn alle möglichen wissenschaftlichen Fragen beantwortet sind, unsere Lebensprobleme noch gar nicht berührt sind. Freilich bleibt dann eben keine Frage mehr; und eben dies ist die Antwort.*”

Ludwig Wittgenstein,
Tractatus logico-philosophicus, Logisch-philosophische Abhandlung;
Suhrkamp, Frankfurt a. M., 2003

Abstract

The relativistic pseudopotential (PP) method is one of the most common and successful approximations in computational quantum chemistry. If suitably parameterized – e.g., fitted to atomic valence total energies from highly accurate relativistic reference calculations –, atomic PPs provide effective (spin–orbit) 1-electron operators mimicking the chemically inert atomic core subsystem, which thus is excluded from explicit considerations.

This work deals with the development of a Kramers-restricted, 2-component PP Hartree–Fock SCF program based on the spin-restricted, 1-component HF SCF modules of the “Quantum Objects Library” of C++ program modules at the Dolg and Hanrath groups at Cologne University. Kramers’ restriction, i.e. time reversal symmetry, is addressed at the lowest hierarchical level of the (formally complexified) matrix algebra modules. PP matrix elements are computed using PP integral subroutines of the ARGOS program, which are interfaced to the existing structure. On this basis, a set of spin-restricted, 1-component (all-electron and) spin-free PP, and Kramers-restricted, 2-component spin–orbit PP HF SCF programs is implemented. “Optimal damping” and initial guess density matrices constructed from atomic densities are shown to improve SCF convergence significantly. As first steps towards correlated 2-component calculation schemes, a modular structure for matrix–matrix multiplication-driven 4-index integral transformations to the Fockian eigenbasis is developed, and preliminary 2-component MP2 calculations are presented.

Kurzzusammenfassung

Eines der am weitesten verbreiteten und erfolgreichsten Näherungsverfahren der computergestützten Quantenchemie ist das der relativistischen Pseudopotenziale (PP). Geeignete parametrisierte PPs, die etwa durch Fits an atomare Valenzgesamtenergien aus hochgenauen relativistischen Referenzrechnungen erhalten werden können, stellen effektive (spin- und bahndrehimpulsabhängige) Einelektronenoperatoren dar, die den chemisch inerten Atomrumpf simulieren. Letzterer wird somit von der expliziten Betrachtung ausgeschlossen.

Gegenstand dieser Arbeit ist die Entwicklung eines Kramers-eingeschränkten, 2-komponentigen PP-Hartree–Fock SCF-Programms auf Grundlage der spin-eingeschränkten, 1-komponentigen Module der “Quantum Objects Library”-Bibliothek von C++-Programmmodulen in den Arbeitsgruppen Dolg und Hanrath an der Universität zu Köln. Die Kramers-Beschränkung, d.h. die Forderung nach Invarianz bezüglich Zeitumkehr, wird auf der hierarchisch tiefsten Ebene der (formal komplexifizierten) Matrixalgebra-Module realisiert. Zur Berechnung von PP-Matrixelementen werden Teile des ARGOS-Programms in die vorhandene Struktur integriert. Auf dieser Basis werden spin-eingeschränkte, 1-komponentige “*all-electron*”- und PP-, sowie Kramers-eingeschränkte, 2-komponentige spin–bahngekoppelte PP-HF SCF-Programme implementiert. Es wird gezeigt, dass sowohl die Methode der “optimalen Dämpfung”, als auch die Verwendung von aus atomaren Dichtematrizen konstruierten “*initial guess*”-Dichtematrizen die Konvergenz des SCF-Verfahrens bedeutend verbessern. Weiterführende Schritte zu 2-komponentigen korrelierten Verfahren beinhalten die Entwicklung einer modularen Programmstruktur zur Transformation der 4-Index-Integrale auf die Eigenbasis des Fock-Operators unter Ausnutzung schneller Matrix–Matrix-Multiplikation. Abschließend werden erste 2-komponentige MP2-Rechnungen vorgestellt.

Contents

Abstract, Kurzzusammenfassung	iv
1 Introduction	1
2 Principles of Relativistic Quantum Chemistry	4
2.1 Equation of Motion, Time, and Time Reversal	5
2.1.1 Time-Dependent and -Independent Schrödinger Equations	5
2.1.2 Time Reversal in Quantum Mechanics	6
2.2 The Dirac Hydrogen Atom	10
2.3 Approximations to The Dirac Equation	14
2.3.1 Small Component Elimination: Pauli-Like Theory	15
2.3.2 Semilocal Atomic Pseudopotential Approximations	16
3 2-Component HF SCF Theory	21
3.1 Kramers-Restricted 2-Component HF SCF Theory	22
3.1.1 The Slater Determinant Ansatz	22
3.1.2 Nature and Definition of The 1-Electron 2-Spinors	24
3.1.3 Kramers' Restriction and Time Reversal Symmetry	26
3.2 Kramers-Restricted Roothaan–Hall Equations	29
3.2.1 Expansion in Terms of Scalar Basis Functions	30
3.2.2 Time Reversal Invariance and Matrix Symmetry	31
3.2.3 Spin Component Decoupling in the $\hat{B} \rightarrow \hat{0}$ Limit	35
3.2.4 Cartesian Gaussian-Type Orbital Basis Functions	35
3.3 Closing Comments on Correlation	36

4 Scope, Strategy, Status Quo 2006	39
4.1 The “Quantum Objects Library”	39
4.2 Scope, Strategy, and Organization of This Work	40
4.2.1 Implementation Goals	40
4.2.2 Implementation Strategy and Outline	41
4.3 Quantum Objects Library: Status Quo End 2006	43
4.3.1 QOL Matrix and Matrix Representation Classes	43
4.3.2 QOL Iteration, Integration, and Integral Communication .	46
4.3.3 The QOL Self-Consistent Field Algorithm	53
5 Matrix Algebra	55
5.1 Matrix Container and Algebra Classes	56
5.1.1 Hermitian and Time Reversal-Like Block Symmetry . . .	56
5.1.2 Quaternion Matrix Matrix Representations	59
5.2 Matrix Representation Classes	60
5.2.1 QOL Unitary Spaces and Scalar Products	61
5.2.2 Symmetric, Hermitian, and Hermitian Time Reversal-Invariant Matrix Representations	65
5.3 Output Formatting and Visualization	67
6 2-Component Pseudopotentials	70
6.1 General Design Principles	71
6.2 Pseudopotential Definition and Classes	73
6.2.1 Input and Parameter Definition	73
6.2.2 Modifications to the Atom and Molecule Classes: Parsing	75
6.2.3 QOL 2-Component Semilocal Atomic Pseudopotentials .	77
6.3 Pseudopotential Integrals	79
6.3.1 ARGOS Integral Subroutines: Overview	80
6.3.2 ARGOS–QOL Interfacing	82
6.3.3 Integration and Pseudopotential Integral Communication .	86
7 Self-Consistent 2-Spinor Fields	90
7.1 Spin Component Exchange Coupling	90
7.2 Eigenvector and Density Matrix Processing	95

7.2.1	Top-Level Self-Consistent Field Algorithm Classes	95
7.2.2	Choosing Time Reversal-Invariant Eigenspace Bases	97
7.2.3	Atomic Occupation Numbers and Angular Symmetry	102
7.3	Optimal Damping	107
7.4	Molecule-From-Atoms Initial Guess Densities	112
8	4-Index Integral Transformation and MP2	116
8.1	Preliminary Considerations	118
8.2	Integral Transformation to Fockian Eigenbases	119
8.2.1	Broken Spin Symmetry: Index Pair Transformation	119
8.2.2	A Naïve 4-Index Integral Transformation Algorithm	121
8.2.3	BLAS3 Multiplication-Driven Auxiliary and Matrix Classes	123
8.3	Kramers-Restricted Møller-Plesset Perturbation Theory	127
9	Summary and Outlook	132
Bibliography		137
List of Figures		147
List of Tables		148
Acronyms and Abbreviations		149
Erklärung gemäß §4 Abs. 1, 9. der Promotionsordnung		151
Curriculum Vitae		152
Acknowledgments		153

1

Introduction

The term “relativistic effect” is not easily defined. As put in a footnote to a recent paper by Wang et al.,¹

“[The] world is relativistic, the nonrelativistic Schrödinger approach being a reasonable approximation for lighter elements only.”

In this spirit, one could state that relativistic effects in chemistry^{2,3} are discrepancies arising if formally non-relativistic considerations are juxtaposed with observations of nature or, most often, accurate relativistic electronic structure calculations. Thus, increasingly popular statements as, e.g.,⁴ “*Relativistic effects play an important role in the chemistry of [the heaviest main-group elements]*” have to be understood in the sense that non-relativistic quantum chemistry ceases to provide the correct picture.

This is already the case for second- and, to a greater or lesser extent, first-row transition metals:^{2,3,5} Spin-orbit interactions⁶ are decisive for ground state configurations of elements as light as nickel.^{1,7} The situation is even more pronounced for heavy and super-heavy elements^{2,3,8} and, of course, especially important for magnetic properties⁵ and optical spectroscopy.^{6,9}

With the Dirac–Coulomb–Breit (DCB) Hamiltonian, possibly even including corrections due to quantum electrodynamics, today’s machinery of wave function-based correlation methods^{2,3,10,11} allows relativistic *ab initio* calculations of atomic and small molecular systems with remarkable accuracy. However, 4-component

DCB schemes are generally not affordable for systems of chemical interest, and often are also not mandatory: In fact, chemistry and large parts of molecular physics are dominated by comparably small energies and electronic momenta. Beginning maybe in the late 1980s, a wealth of approximate^{12–15} and “exact” 2-component theories^{16–20} to the Dirac equation has been developed.^{2,10,11} All of these allow to focus the full computational effort to the Dirac spinors’ positive energy components that dominate atomic and molecular electronic structure, and address spin-orbit coupling non-perturbatively from the beginning. Although the “*four components good, two components bad!*” debate²¹ has not been settled, it is safe to say that modern 2-component methods have been proven to be both affordable and reliable tools for relativistic quantum chemistry.

The relativistic pseudopotential approximation^{2,22–24} is conceptually different in the sense that it does not aim at the decoupling of the Dirac equation directly. Instead, it provides a set of effective, variationally stable 1-electron (spin-orbit) operators^{25,26}

$$\hat{W} = \hat{W}_{LJ} + \sum_{l=0}^{L-1} \sum_{m_l} \hat{A}_l |lm_l\rangle\langle lm_l| + \hat{\mathbf{S}} \cdot \sum_{l=0}^{L-1} \sum_{m_lm'_l} \hat{B}_l |lm_l\rangle\langle lm_l| \hat{\mathbf{L}} |lm'_l\rangle\langle lm'_l|$$

that allow to solve electronic Schrödinger equations for valence electrons only: The excluded atomic core subsystems – for which relativistic considerations are generally most important – are mimicked by spin- and orbital angular momentum-dependent “pseudo-potentials” obtained from, e.g., relativistic atomic reference calculations.²⁷

The aim of this work is the development of a Kramers-restricted 2-component pseudopotential Hartree–Fock self-consistent field program^{28–30} on the basis of the spin-restricted HF SCF parts of Cologne’s “Quantum Objects Library” set of C++ program modules. From their intimate connection to the QOL’s larger parts, providing well-developed (arbitrary excitation single- and multi-reference) Coupled Cluster modules, this is expected to give way to the possibility to perform high-level correlated electronic structure calculations also at the relativistic 2-component level in the foreseeable future.

The adopted bottom-up implementation strategy, discussed in detail in ch. 4.3, reflects, to a large extent, the organization of this work:

In 2-component Roothaan–Hall HF SCF theory, operator matrix representations have twice the row and column dimensions as compared to the 1-component, spin-restricted case, and are generally complex-valued. Kramers’ restriction, i.e. invariance with respect to time inversion, manifests in special matrix block symmetries. The necessary modifications and extensions of the QOL’s matrix algebra modules are discussed in ch. 5. 2-component pseudopotential integrals over Cartesian Gaussian-Type Orbital basis functions^{31,32} are calculated using the ARGOS integral program^{31–35} of Pitzer et al., which has been interfaced to the QOL as described in ch. 7. 6.

The 1-component spin-restricted all-electron and spin-free PP, and 2-component Kramers-restricted spin-orbit PP Hartree–Fock SCF programs are presented in ch. 7; all have been supplemented to allow fractional atomic occupation numbers, “optimal damping”^{36,37}, and improved SCF initial guesses.³⁸

On the basis of these programs, ch. 8 presents design and implementation of modules for 4-index integral transformation to the molecular 2-spinor basis, and preliminary correlated calculations at a 2-component Møller–Plesset perturbation theory level to second order.

2

Principles of Relativistic Quantum Chemistry

The purpose of this first, introductory chapter is to provide a brief overview of the larger context, the underlying concepts, and a number of more special issues of central importance for this work. In sec. 2.1 the Schrödinger equation of motion, the special role of time in quantum mechanics, and – in more detail – the symmetry operation of time reversal are introduced. Then, sec.s 2.2 and 2.3 outline the most basic features of relativistic quantum chemistry for the particularly simple example of the Dirac equation for the hydrogen atom – which eases the discussion of angular momentum and spin–orbit coupling from the spherical symmetry of the potential –, and of 2-component approximation schemes to this Dirac equation, respectively; the 2-component pseudopotential approximation is discussed in more detail in sec. 2.3.2.

It is clear that the presentation of this chapter cannot be comprehensive by any means. Instead, the discussion has been given an operational focus with the particular aim of fixing notation and introducing important relations as reference for the following parts of this work.

2.1 Equation of Motion, Time, and Time Reversal

The following discussion gives brief accounts of the time-dependent and -independent Schrödinger equations in sec. 2.1.1 mainly to provide grounds for the detailed discussion of the symmetry operation of time reversal and its properties in sec. 2.1.2.

2.1.1 Time-Dependent and -Independent Schrödinger Equations

Central to almost all wave function-based quantum theories of atomic and molecular physics and chemistry – independent of the nature, number, and types of interactions among the system of interest’s particles – is the time-dependent Schrödinger equation^{39–41a}

$$i \frac{d}{dt} |\Psi_t\rangle = \hat{E}_t |\Psi_t\rangle. \quad (2-1)$$

Eq. 2-1 defines both the system and its state by its hermitian “Hamiltonian” energy operator \hat{E} and state function $|\Psi_t\rangle$, respectively. Writing⁴¹

$$\frac{d}{dt} |\Psi_t\rangle := \lim_{\tau \rightarrow 0} \frac{|\Psi_{t+\tau}\rangle - |\Psi_t\rangle}{\tau} \quad (2-2)$$

$$\Rightarrow |\Psi_{t+dt}\rangle = (\hat{1} - i\hat{E}dt)|\Psi_t\rangle =: \hat{U}_{t+dt}|\Psi_t\rangle \quad (2-3)$$

defines the infinitesimal unitary “time translation” operator \hat{U}_{t+dt} that effects propagation of the state function from time t to time $t + dt$. For the special, but common case of time-independent Hamiltonians \hat{E} , the state $|\Psi_t\rangle$ at time t is thus defined by the finite transformation

$$|\Psi_t\rangle = e^{-i\hat{E}t} |\Psi_0\rangle = \hat{U}_t |\Psi_t\rangle. \quad (2-4)$$

The $t = 0$ state $|\Psi_0\rangle$ is formally the single integration constant for the solution of eq. 2-1 with $\hat{E}_t = \hat{E}$.^b

^a The primary reference is probably Schrödinger’s “*Undulatory Theory of the Mechanics of Atoms and Molecules*”,³⁹ but central ideas have been published elsewhere.^{42–45} Here and in the following, the term “Schrödinger equation” refers to any equation of the form 2-1, with \hat{E}_t not necessarily restricted to Schrödinger’s non-relativistic energy operator.

^b There is no (hermitian) time operator or observable in quantum mechanics.⁴⁶ Instead, t is understood as parameter that formally labels a family $(|\Psi_t\rangle)_{t \in \mathbb{R}}$ representing propagation of the

For the stationary states considered in the largest part of this work, $|\Psi_t\rangle$ of eq. 2-4 is an eigenfunction of \hat{E} . Then, the time dependency can be factored out as time-dependent phase e^{-iEt} , and eq. 2-1 simplifies to^{40,41}

$$\hat{E}|\Psi\rangle = \mathcal{E}|\Psi\rangle \quad \text{with} \quad |\Psi_t\rangle = e^{-i\mathcal{E}t}|\Psi\rangle. \quad (2-5)$$

2.1.2 Time Reversal in Quantum Mechanics

“ T symmetry”, i.e. the invariance of (a subset of) the laws of physics under time reversal^{46–50}

$$\mathcal{T} : t \mapsto -t, \quad (2-6)$$

is a fundamental discrete symmetry of systems of chemical interest. T symmetry can, to some extent, be exploited similarly to point and space group symmetries to give insight to a given system’s physics, and to simplify its abstract description.^{49,51,52} Put somewhat simplified, if for a time reversal-invariant system a given trajectory is accessible, T symmetry allows also the reversed trajectory.

It is stressed that “time reversal” does not refer to “going backwards in time”, but is best understood as “motion reversal”:^{46,47,49} \mathcal{T} transforms a given system’s dynamical variables of position and (conjugate) momentum as

$$\mathbf{r} \mapsto \mathbf{r}, \quad \mathbf{p} \mapsto -\mathbf{p}; \quad (2-7)$$

consequently, $\mathbf{L} \mapsto -\mathbf{L}$.

Within the contemporary experimental uncertainty there is no evidence that electromagnetic interactions break T symmetry.^c Therefore, the relativistic elec-

state. Moreover, the “time derivative” of eq. 2-1 is not an operator on the state Hilbert space \mathbb{E} : Eq. 2-1 is, more precisely, to be understood as a parametrization of $(|\Psi_t\rangle)_{t \in \mathbb{R}} \subset \mathbb{E}$ such that $\hat{E}_t : \mathbb{E} \rightarrow \mathbb{E}$, acting on $|\Psi_t\rangle$, is identical to the $\tau \rightarrow 0$ limit of the difference quotient of eq. 2-2.⁴⁶ Therefore, time dependency of both operators and states is indicated by a subscript t , e.g. $|\Psi_t\rangle$, instead of by the possibly more suggestive notation $|\Psi(t)\rangle$.

^c Note that nature does break T symmetry. If the CPT theorem^{49,53} is true, CP violations – as experimentally observed in, e.g., kaon and B meson decay^{54,55} – imply T violation. Note, too, that the collective behavior of macroscopic ensembles does display a “time direction”, seemingly contrasting time reversal invariance of the systems’ equations of motion. However, whereas T violation by the weak interaction is truly a consequence of the symmetry properties of the system dynamics, macroscopic irreversibility is a purely statistical phenomenon independent of time or time reversal.^{49,56}

tromagnetic Hamiltonian \hat{E}_t of eq. 2-1 is assumed to be time reversal-invariant in the absence of external magnetic fields.^d

In quantum mechanics, time reversal is different from the $t \mapsto -t$ operation 2-6 because of the special role of the time variable t . The time-reversal transformed state $|\Psi_t\rangle$ is

$$\hat{k}|\Psi_{-t}\rangle = |\bar{\Psi}_{-t}\rangle, \quad (2-8)$$

i.e. obtained from $|\Psi_t\rangle$ by both the \mathcal{T} operation 2-6 and action of Wigner's anti-unitary time reversal operator \hat{k} .⁴⁷⁻⁴⁹ For consistency with eq.s 2-7

$$\begin{aligned} \langle \mathbf{r} \rangle_{\Psi_t} \mapsto \langle \mathbf{r} \rangle_{\bar{\Psi}_{-t}} &= \langle \Psi_{-t} | \hat{k}^\dagger | \hat{\mathbf{r}} \hat{k} \Psi_{-t} \rangle \Rightarrow \hat{k} \hat{\mathbf{r}} \hat{k}^\dagger = \hat{\mathbf{r}}, \\ \langle \mathbf{p} \rangle_{\Psi_t} \mapsto -\langle \mathbf{p} \rangle_{\bar{\Psi}_{-t}} &= -\langle \Psi_{-t} | \hat{k}^\dagger | \hat{\mathbf{p}} \hat{k} \Psi_{-t} \rangle \Rightarrow \hat{k} \hat{\mathbf{p}} \hat{k}^\dagger = -\hat{\mathbf{p}}; \end{aligned} \quad (2-9)$$

$$\Rightarrow \hat{k} i \delta_{xy} \hat{k}^\dagger = \hat{k} [\hat{x}, \hat{p}_y]_- \hat{k}^\dagger = [\hat{k} \hat{x} \hat{k}^\dagger, \hat{k} \hat{p}_y \hat{k}^\dagger]_- = -i \delta_{xy} \hat{1} \quad (2-10)$$

and similar for all pairs of Cartesian coordinates x, y, z , because eq.s 2-9 must hold component-wise. From eq. 2-10 \hat{k} is anti-linear,^{47,57} i.e. \hat{k} effects complex conjugation of numbers by conjugation. A number of special algebraic properties of \hat{k} and its explicit form (in the position representation) are discussed in more detail below, but its physical interpretation is clear from eq.s 2-9 and 2-10:

If $|\Psi_t\rangle$ is a solution of eq. 2-1, i.e.

$$i \frac{d}{dt} |\Psi_t\rangle = \hat{E} |\Psi_t\rangle \quad (2-11)$$

$$\Rightarrow \hat{k} i \frac{d}{dt} |\Psi_t\rangle = -i \frac{d}{dt} |\hat{k} \Psi_t\rangle =: -i \frac{d}{dt} |\bar{\Psi}_{-t}\rangle = \hat{E} |\bar{\Psi}_{-t}\rangle \quad (2-12)$$

provided $\hat{E} = \hat{k} \hat{E} \hat{k}^\dagger$; then, applying \mathcal{T} : $t \mapsto -t$ shows^{47,49}

$$i \frac{d}{dt} |\bar{\Psi}_{-t}\rangle = \hat{E} |\bar{\Psi}_{-t}\rangle. \quad (2-13)$$

Thus, if $|\Psi_t\rangle$ is a solution of the time-dependent Schrödinger equation 2-1, then $|\bar{\Psi}_{-t}\rangle$ is a solution of the same equation of motion, obtained from the first solution

^d The term “external” has a precise meaning in this context: If the field $\mathbf{B} = \nabla \times \mathbf{A}$, defined by its vector potential \mathbf{A} , is provided by system components also subject to the \mathcal{T} transformation, $\mathbf{A} \mapsto -\mathbf{A}$, and \hat{E}_t is time reversal-invariant; “external” thus refers to field sources not explicitly addressed in terms of field source dynamical variables by eq. 2-1, i.e. not simultaneously reversed under \mathcal{T} .⁵¹

at time $-t$ by the transformation \hat{k} . Note that the “time reversal operator” \hat{k} is distinct from time reversal \mathcal{T} ; an anti-unitary operator cannot act non-trivially on a real parameter.

Brief Review of Some Properties of Anti-Linear and Anti-Unitary Operators

Anti-linear and, thus, anti-unitary operators behave in a slightly different way as compared to linear and unitary operators typically employed in quantum mechanics.^{47,51,57–59} Only a brief account of the operationally most important manipulation rules is given here, mainly to fix notations and provide a reference for the rest of this work.

- Anti-linear operators \hat{k} on Hilbert spaces \mathbb{E} act, for $C_a, C_b \in \mathbb{C}$ and $|\psi_a\rangle, |\psi_b\rangle \in \mathbb{E}$, as

$$\hat{k}(C_a|\psi_a\rangle + C_b|\psi_b\rangle) = C_a^*|\hat{k}\psi_a\rangle + C_b^*|\hat{k}\psi_b\rangle. \quad (2-14)$$

Note that it is necessary to explicitly indicate whether \hat{k} operates on the anti-linear or linear, i.e. on the bra or ket argument of the inner product; formal expressions as, e.g., $\langle\psi_a|\hat{k}|\psi_b\rangle$ are ambiguous.⁴⁷

- As \hat{k}^\dagger is also anti-linear, \hat{k}^\dagger is defined by

$$\langle\psi_a|\hat{k}^\dagger|\psi_b\rangle := \langle\psi_a|\hat{k}\psi_b\rangle^* = \langle\psi_b|\hat{k}^\dagger|\psi_a\rangle. \quad (2-15)$$

- Anti-unitary, i.e. anti-linear unitary operators \hat{k} preserve the (positive-semi-definite) norm $\|\psi_a\|^2 = \|\hat{k}\psi_a\|^2$, but

$$\hat{k}^\dagger = \hat{k}^{-1} \quad (2-16)$$

$$\Rightarrow \langle\psi_a|\hat{k}^\dagger|\hat{k}\psi_b\rangle = \langle\psi_a|\hat{k}^\dagger\hat{k}\psi_b\rangle^* = \langle\psi_a|\psi_b\rangle^* = \langle\psi_b|\psi_a\rangle. \quad (2-17)$$

The operator \hat{C}_B of complex conjugation is a special anti-linear operator that is defined in terms of a basis B only.^{47,57} For a given B , \hat{C}_B is the anti-linear operator that leaves all $|\phi_p\rangle \in B$ invariant. If $\langle\phi_p|\phi_q\rangle = \delta_{pq}$, \hat{C}_B is anti-unitary;⁵⁷ moreover, $\hat{C}_B^2 = \hat{1}$ for all B . Messiah⁴⁷ provides a comprehensive discussion of changes of bases.

The Time Reversal Operator and Kramers' Theorem

An abstract, representation-independent definition of the time reversal operator \hat{k} can only be given in terms of eq.s 2-9, i.e. the time reversal transformation of the dynamical variables $\hat{\mathbf{r}}$ and $\hat{\mathbf{p}}$; for quantum systems with spin, eq.s 2-9 have to be supplemented by

$$\hat{k}\hat{\mathbf{S}}\hat{k}^\dagger = -\hat{\mathbf{S}} \quad (2-18)$$

as the total angular momentum $\hat{\mathbf{J}} = \hat{\mathbf{L}} + \hat{\mathbf{S}}$ must transform consistently with $\hat{\mathbf{L}} = \hat{\mathbf{r}} \times \hat{\mathbf{p}} \Rightarrow \hat{k}\hat{\mathbf{L}}\hat{k}^\dagger = -\hat{\mathbf{L}}$.^e

In the position basis and, for the spin variable, the basis $(|\alpha\rangle, |\beta\rangle)$ of \hat{S}_z eigenfunctions, for a single electron⁴⁷⁻⁴⁹

$$\hat{k} := -i2\hat{S}_y\hat{\mathcal{C}}, \quad (2-19)$$

$$\hat{k}: |\alpha\rangle \mapsto |\hat{k}\alpha\rangle = -|\beta\rangle, \quad \hat{k}: |\beta\rangle \mapsto |\hat{k}\beta\rangle = |\alpha\rangle. \quad (2-20)$$

$\hat{\mathcal{C}} := \hat{\mathcal{C}}_{(|\mathbf{r}\rangle, |\alpha\rangle, |\beta\rangle)}$, is the anti-linear complex conjugation operator in the $(|\mathbf{r}\rangle, |\alpha\rangle, |\beta\rangle)$ basis; as, in this basis, \hat{S}_x and \hat{S}_z are purely imaginary and $\hat{\mathcal{C}}$ thus effects sign change, eq. 2-18 requires an additional unitary transformation $e^{-i\pi\hat{\mathbf{S}} \cdot \mathbf{e}_y} = -i2\hat{S}_y$ on the spin- $\frac{1}{2}$ Hilbert space, i.e. a spin rotation about the y axis \mathbf{e}_y by π .

From eq. 2-19

$$\hat{k}\hat{k} = \hat{k}^2 = -\hat{1}; \quad (2-21)$$

as $\hat{\mathcal{C}}^2 = \hat{1}$, this can essentially be backtraced to the spin space rotation by $\pi + \pi$, i.e. the special spinor transformation behavior under rotation. Because of eq.s 2-17, 2-21, and 2-15,

$$\langle \Psi | \hat{k} \Psi \rangle = \langle \Psi | \hat{k}^\dagger | \hat{k} \hat{k} \Psi \rangle^* = -\langle \Psi | \hat{k}^\dagger | \Psi \rangle^* = -\langle \Psi | \hat{k} \Psi \rangle = 0 \quad (2-22)$$

shows that \hat{k} has no eigenfunctions and, thus, no spectral decomposition.^{47,49,51} As the time reversal operator $\hat{\mathcal{K}}$ for an N -electron system is simply the product operator

$$\hat{\mathcal{K}} := \hat{k}_1 \otimes \cdots \otimes \hat{k}_i \otimes \cdots \otimes \hat{k}_N \quad (2-23)$$

$$\Rightarrow \quad \hat{\mathcal{K}}^2 = (-\hat{1})^N, \quad (2-24)$$

^e With eq. 2-18, \hat{k} commutes with all rotations of position and/or spin space:⁴⁷ Because \hat{k} is anti-unitary, $\hat{k}e^{-i\hat{\mathbf{J}} \cdot \mathbf{n}\varphi}\hat{k}^\dagger = e^{i\hat{k}\hat{\mathbf{J}} \cdot \mathbf{n}\hat{k}^\dagger\varphi} = e^{-i\hat{\mathbf{J}} \cdot \mathbf{n}\varphi}$. In fact, \hat{k} commutes with all operators of spatial symmetry transformations, i.e. translations, rotations, and reflexions.⁴⁷

eq. 2-22 is also valid for any N -electron system with N odd.

Eq.s 2-23 and 2-24 allow the algebraic proof^{47–49} of Kramers' theorem,⁶⁰ i.e. of the theorem that, in the absence of external magnetic fields, all energy levels of a system of odd numbers of electrons are at least two-fold degenerate; and, in fact, every such degeneracy is even-fold. Clearly, if $\hat{E}|\Psi\rangle = \mathcal{E}|\Psi\rangle$, then $\hat{E}|\hat{\mathcal{K}}\Psi\rangle = \hat{\mathcal{K}}\mathcal{E}|\Psi\rangle = \mathcal{E}\hat{\mathcal{K}}|\Psi\rangle$ because \hat{E} is time reversal-invariant; but, as $\langle\Psi|\hat{\mathcal{K}}^\dagger|\Psi\rangle = 0$ for all $|\Psi\rangle$, $|\Psi\rangle$ and $|\hat{\mathcal{K}}\Psi\rangle$ are independent states.

Because of eq. 2-21 and, in consequence, eq. 2-24, an N -electron state function cannot be made invariant under $\hat{\mathcal{K}}$. However, as $\hat{\mathcal{K}}^2 = \hat{1}$ for N even, one can always choose a phase factor for the N -electron state $|\Psi\rangle$ such that $|\hat{\mathcal{K}}\Psi\rangle = |\Psi\rangle$,⁵¹ as exploited in the context of, e.g., Kramers-restricted Hartree–Fock SCF theory^{28,29} discussed in detail in ch. 3.

2.2 The Dirac Hydrogen Atom

The preceeding discussion of sec. 2.1 has been set up very general and did not refer to the precise nature of the Hamiltonian energy operator \hat{E} of, e.g., eq. 2-1. The present section is focussed on the relativistic Hamiltonian and state function for a single electron only, and will address only a number of points in detail that provide the basis for the following parts.^f

For a single relativistic electron in the proton's time-independent electrostatic potential \hat{V} , the Hamiltonian \hat{E} of eq. 2-1 is the Dirac operator^{10,11,61}

$$\hat{h}_D = c_0 \hat{\alpha} \cdot \hat{\mathbf{p}} + \hat{\beta} m_e c_0^2 + \hat{V} \quad (2-25)$$

with c_0 the vacuum speed of light; the electron rest mass $m_e = 1$ a.u. has been written explicitly for clarity.^g

^f Whereas the restriction to the spherical, point-like proton Coulomb potential is, of course, a limitation from the point of view of a general (molecular) electrostatic potential, the system's angular symmetry allows to clarify particularly the coupling of orbital angular momentum and spin, which is of central importance.

^g Here and in the following, the electron–proton interaction is assumed to be instantaneous, i.e. retardation and QED effects are neglected.¹¹ Moreover, the Born–Oppenheimer approximation is applied, and the proton's spin and vector potential \mathbf{A} are omitted.

Eq. 2-25 cannot be “derived” or motivated in any detail here. It must suffice to briefly address the key points:

To arrive at a relativistic, Lorentz-invariant equation of motion of first order in spatial and time variables, the square root argument of the relativistic energy-momentum relation⁶²

$$E = c_0 \sqrt{m_e^2 c_0^2 + \mathbf{p} \cdot \mathbf{p}} + V \quad (2-26)$$

is assumed to be a perfect square $(\boldsymbol{\alpha} \cdot \mathbf{p} + \beta c_0 m_e)^2$, the quantities $\alpha_x, \alpha_y, \alpha_z$, and β are to be defined. Then, eq. 2-1 with $\hat{E} = \hat{h}_D$ of eq. 2-25 follows immediately from the “correspondence principle”. The algebra of the components of the $\hat{\boldsymbol{\alpha}}$ vector and $\hat{\beta}$ operator follow from the requirement to match eq. 2-26, i.e.

$$m_e^2 c_0^2 + \hat{\mathbf{p}} \cdot \hat{\mathbf{p}} \stackrel{!}{=} \hat{\boldsymbol{\alpha}} \cdot \hat{\mathbf{p}} + \hat{\beta} c_0 m_e \quad (2-27)$$

$$\Rightarrow \hat{\alpha}_r^2 = \hat{\beta}^2 = \hat{1}, \quad [\hat{\alpha}_r \hat{\alpha}_s]_+ = \delta_{rs} \hat{1}, \quad [\hat{\alpha}_r, \hat{\beta}]_+ = \hat{0} \quad (2-28)$$

for Cartesian coordinates $r, s \in \{x, y, z\}$. In the Dirac “standard” representation, the $\hat{\alpha}_r$ and $\hat{\beta}$ operators are represented as 2×2 matrices

$$\hat{\alpha}_r \equiv \begin{pmatrix} \hat{\alpha}_r^{++} & \hat{\alpha}_r^{+-} \\ \hat{\alpha}_r^{-+} & \hat{\alpha}_r^{--} \end{pmatrix} = \begin{pmatrix} \hat{0} & \hat{\sigma}_r \\ \hat{\sigma}_r & \hat{0} \end{pmatrix}, \quad \hat{\beta} \equiv \begin{pmatrix} \hat{1} & \hat{0} \\ \hat{0} & -\hat{1} \end{pmatrix} \quad (2-29)$$

in terms of the Pauli sigma operators $\hat{\sigma}_r$.^h Consequently, the solution of the Dirac equation is a quantity

$$|\Psi_t\rangle \equiv (|\Psi_t^+\rangle, |\Psi_t^-\rangle)^T, \quad (2-30)$$

the 2-spinors $|\Psi_t^+\rangle$ and $|\Psi_t^-\rangle$ being referred to as “large” and “small” components, respectively.

In a given frame of reference, which is mostly the Born–Oppenheimer frame, the time-dependence of the stationary 1-electron state $|\Psi_t\rangle$ can be factored out to give the time-independent Dirac equation.^{10,11} With eq. 2-29,

$$(\hat{V} + m_e c_0^2) |\Psi^+\rangle + c_0 \hat{\boldsymbol{\sigma}} \cdot \hat{\mathbf{p}} |\Psi^-\rangle = E |\Psi^+\rangle, \quad (2-31)$$

$$c_0 \hat{\boldsymbol{\sigma}} \cdot \hat{\mathbf{p}} |\Psi^+\rangle + (\hat{V} - m_e c_0^2) |\Psi^-\rangle = E |\Psi^-\rangle \quad (2-32)$$

^h The Dirac standard representation corresponds to a choice of $(1, 0)^T$ and $(0, 1)^T$ as vector representations of the “positive” and “negative energy basis functions” $|+\rangle$ and $|-\rangle$, respectively. Consequently, the standard representation of eq. 2-29 is defined up to a unitary transformation,^{11,63} but only eq. 2-29 will be referred to here.

where the vector operator $\hat{\sigma}$ collects the Pauli sigma operators $\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z$.

No detailed discussion of the solutions or the properties of the solutions of eq.s 2-31 and 2-32 will be given at this point. Instead, only a number of selected points are addressed that aim at the following discussion of (2-component spin-orbit pseudopotential) approximations to the Dirac equation.

Notes on The Dirac Hamiltonian Spectrum and The 4-Spinor Structure

The Dirac Hamiltonian \hat{h}_D of eq. 2-25 allows negative energy eigenvalues E and, moreover, is not bounded from below. Instead, the \hat{h}_D spectrum has continua for $E \geq m_e c_0^2$ and $E \leq -m_e c_0^2$, as well as a number of discrete bound states in the interval $[0, m_e c_0^2]$.^{10,11}

The existence of positive and negative energy solutions is directly connected to the 4-spinor nature of the state function $|\Psi\rangle$, i.e. the presence of both large and small components $|\Psi^+\rangle$ and $|\Psi^-\rangle$, whereas a direct physical interpretation is difficult. As can be seen from eq.s 2-31 and 2-32, $|\Psi^+\rangle$ and $|\Psi^-\rangle$ are coupled by the off-diagonal operators $\hat{h}_D^{+-} = \hat{h}_D^{-+} = c_0 \hat{\sigma} \cdot \hat{\mathbf{p}}$. Operationally, for $E - m_e c_0^2 \approx 0$ this coupling is “small” in the sense that

$$\|\Psi^+\|^2 \gg \|\Psi^-\|^2,$$

i.e. that the state function $|\Psi\rangle$ is dominated by the large component, justifying the “large” label.ⁱ

The observation that $|\Psi\rangle$ is dominated by $|\Psi^+\rangle$ – with the notable and, quantitatively, important exception of inner-shell electrons of heavy and super-heavy

ⁱ It is noted in passing that for large electron velocities, i.e. for very strongly bound and high-energy continuum states with $E \approx 0$ and $E \gg m_e c_0^2$, respectively, the large component is generally not large in the stated sense. Moreover, for negative energy states generally $\|\Psi^-\|^2 \gg \|\Psi^+\|^2$.

The physical interpretation of the negative energy states is, from only the point of view of eq.s 2-31 and 2-32, problematic also within the picture of hole theory,⁶³ i.e. assuming a “Dirac sea” of occupied positronic states.^{63–65} In fact, the Dirac equation’s structure and properties point to the necessity of (Dirac spinor) field quantization,⁶³ i.e. many-particle (-electron and -positron) theories also for a single relativistic free or bound electron.

From the point of this work the significance of the $E < 0$ solutions comes from the fact that, generally, the positive energy functions only cannot span the complete function space \hat{h}_D is defined on; any 1-electron state function $|\Psi\rangle$ has both positive and (most often comparably small) negative energy contributions.

elements – is the point of departure for a number of large component-only approximation schemes.^{10,11j}

Angular Symmetry and Spin–Orbit Coupling in The Dirac Hydrogen Atom

The Dirac equation introduces spin to quantum mechanics in a non-heuristic way, i.e. via the algebra of the $\hat{\alpha}_r$ operators. In fact, the velocity operator $c_0\hat{\alpha}$ does not only couple the large and small components but, within each component, electron orbital angular momentum and spin via

$$c_0\hat{\sigma} \cdot \hat{\mathbf{p}} = c_0|\hat{\mathbf{r}}|^{-2}(\hat{\sigma} \cdot \mathbf{r})(2i\hat{\mathbf{S}} \cdot \hat{\mathbf{L}} + \hat{\mathbf{r}} \cdot \hat{\mathbf{p}}) \quad (2-33)$$

such that the large and small component's angular parts $|X_{jm_j}^+\rangle$ and $|X_{jm_j}^-\rangle$ are¹¹

$$\langle\theta,\varphi|X_{jm_j}^\pm\rangle = \sum_{m_s} C(j \mp \frac{1}{2}, m_j - m_s, s, m_s; j, m_j) \cdot Y_{j \mp \frac{1}{2}}^{m_j \mp m_s}(\theta, \varphi) |sm_s\rangle;$$

$C(l, m_l, s, m_s; j, m_j)$ are Clebsch–Gordan coefficients, $\langle\theta,\varphi|lm_l\rangle = Y_l^{m_l}(\theta, \varphi)$ spherical harmonics, and $|sm_s\rangle = |\frac{1}{2}, \pm\frac{1}{2}\rangle$ the \hat{S}_z eigenfunctions

$$\begin{aligned} |\frac{1}{2}, +\frac{1}{2}\rangle &=: |\alpha\rangle, & \hat{S}_z|\alpha\rangle &= m_s|\alpha\rangle = +\frac{1}{2}|\alpha\rangle, \\ |\frac{1}{2}, -\frac{1}{2}\rangle &=: |\beta\rangle, & \hat{S}_z|\beta\rangle &= m_s|\beta\rangle = -\frac{1}{2}|\beta\rangle \end{aligned} \quad (2-34)$$

already employed in eq.s 2-20. Both $|X_{jm_j}^\pm\rangle$ and, thus, $|\Psi\rangle$ are eigenfunctions of the total and projected total angular momentum

$$\hat{\mathbf{J}} \cdot \hat{\mathbf{J}}|\Psi\rangle = j(j+1)|\Psi\rangle, \quad \hat{J}_z|\Psi\rangle = m_j|\Psi\rangle. \quad (2-35)$$

However, $|\Psi\rangle$ is not an eigenfunction of \hat{L}_z and \hat{S}_z , because both $|X_{jm_j}^\pm\rangle$ are linear combinations of products of spherical harmonics and spin functions with different $m_l = m_j - m_s$ and m_s .^k

^j Note, however, that “relativistic effects” in chemistry^{4,10,11,66} are not necessarily small if only the small component is small.

^k $|\Psi\rangle$ is also no eigenfunction of $\hat{\mathbf{L}} \cdot \hat{\mathbf{L}}$, whereas the $|X_{jm_j}^\pm\rangle$ are, albeit with eigenvalues $l(l+1)$ of different orbital angular momentum quantum numbers l . Note that spin–orbit coupling qualitatively changes the non-relativistic “orbital” pictures as illustrated by, e.g., Dyall and Fægri¹⁰ and Szabo.⁶⁷

Consequently, the energy¹¹

$$E = E_{n|\kappa|} = m_e c_0^2 \left(1 + \frac{Z c_0^{-1}}{\sqrt{\kappa^2 - Z^2 c_0^{-2} + n - |\kappa|}} \right)^{-1/2}, \quad (2-36)$$

with $\kappa = j + \frac{1}{2}$ and the nuclear charge number Z , depends explicitly on $j = l \pm s$. In other words, the non-relativistic energy level degeneracy with respect to the orbital angular momentum quantum number l is lost.

2.3 Approximations to The Dirac Equation

A number of problems arise with the straightforward application of the 1-electron Dirac Hamiltonian \hat{h}_D of eq. 2-25 in the framework of quantum chemistry: The fact that \hat{h}_D is not bounded from below prohibits a simple variational strategy to be employed for solving eq.s 2-31 and 2-32 (whereas similar, yet more involved techniques^{10,11} can be applied somewhat routinely with contemporary 4-component codes). Moreover, the need to explicitly address small component contributions to the electronic state function causes the computational costs to increase significantly, compared to non-relativistic considerations.

As already pointed out in sec. 2.2, the fact that $|\Psi\rangle$ is typically dominated by the large component $|\Psi^+\rangle$, can be exploited to give rise to a variety of 2-component approximations to the Dirac equation that only consider $|\Psi^+\rangle$ or equivalents thereof; the term “2-component” thus refers to the spin components of the 2-spinor $|\Psi^+\rangle = |\Psi_\alpha^+ \otimes \alpha\rangle + |\Psi_\beta^+ \otimes \beta\rangle$.¹

Common to all such 2-component theories^{10,11} is the (formally exact or approximate) decoupling of eq.s 2-31 and 2-32 by elimination,^{68–70} unitary transformations on the basis of Foldy–Wouthuysen^{14,71} or Douglas–Kroll–Hess parameterizations,^{12,13,15,72} and several matrix techniques.^{16–20m} As the field is vast and continuously expanding, no attempt to review any or all of these approximation schemes in detail is made here; the textbooks by Dyall and Fægri,¹⁰ Reiher and

¹ Moreover, most of these approximations allow separation of spin-independent and -dependent terms and, therefore, provide spin-free, 1-component models by omitting the latter.

^mThe pseudopotential approximation discussed in more detail in sec. 2.3.2 is different in this respect, as it does not aim at a decoupling of the Dirac equation directly.

Wolf,¹¹ and, to some extent, Schwabl⁶³ provide comprehensive elaborations and bibliographies.

Instead, the following discussion is restricted to a somewhat naïve, exemplary small component elimination¹¹ leading to a Pauli-like Hamiltonian \hat{h}_P . Whereas \hat{h}_P is of no practical importance for variational calculation schemes, the Pauli-like Schrödinger equation is instructive from a conceptual point of view, and illustrates a number of important points.

2.3.1 Small Component Elimination: Pauli-Like Theory

For the purpose of the following discussion it is convenient to “shift” the \hat{h}_D spectrum by a constant $m_e c_0^2$, i.e. to set $\hat{h}_D \mapsto \hat{h}_D - m_e c_0^2 \hat{1}$, such that, with $|\Psi^\pm\rangle \mapsto e^{im_e c_0^2 t} |\Psi^\pm\rangle$,

$$\hat{V} |\Psi^+\rangle + c_0 \hat{\sigma} \cdot \hat{\mathbf{p}} |\Psi^-\rangle = E |\Psi^+\rangle, \quad (2-37)$$

$$c_0 \hat{\sigma} \cdot \hat{\mathbf{p}} |\Psi^+\rangle + (\hat{V} - 2m_e c_0^2) |\Psi^-\rangle = E |\Psi^-\rangle. \quad (2-38)$$

Effectively this aligns the Dirac and the non-relativistic Schrödinger spectra to a common $E = 0$ reference, i.e. the bound discrete energy levels lie in the $[-m_e c_0^2, 0)$ interval with the positive energy continuum at $E \geq 0$.

Solving eq. 2-38 for $|\Psi^-\rangle$ and substituting in eq. 2-37 gives¹¹

$$\left(\hat{V} + c_0 \hat{\sigma} \cdot \hat{\mathbf{p}} \frac{1}{2m_e c_0^2} (s_E^{-1} c_0 \hat{\sigma} \cdot \hat{\mathbf{p}}) \right) |\Psi^+\rangle = E |\Psi^+\rangle \quad (2-39)$$

with $s_E := 1 - (\hat{V} - E)/2m_e c_0^2$. Eq. 2-39 is formally exact, the small component being “eliminated” from eq. 2-37.ⁿ Expanding the geometric series s_E^{-1} up to first order and approximating E by its non-relativistic counterpart gives the Pauli-like Schrödinger equation and 2-component Hamiltonian

$$\hat{h}_P |\Psi^+\rangle \approx E |\Psi^+\rangle, \quad (2-40)$$

$$\hat{h}_P := \hat{T} + \hat{V} - \frac{1}{8m_e^3 c_0^2} |\hat{\mathbf{p}}|^4 + \frac{1}{8m_e^2 c_0^2} \Delta \hat{V} + \frac{Z}{2m_e^2 c_0^2} |\hat{\mathbf{r}}|^{-3} \hat{\mathbf{S}} \cdot \hat{\mathbf{L}} \quad (2-41)$$

ⁿ Note that, as $(\hat{\sigma} \cdot \hat{\mathbf{p}}) s_E^{-1} (\hat{\sigma} \cdot \hat{\mathbf{p}}) = \hat{\mathbf{p}} \cdot \hat{\mathbf{p}} + i\hat{\sigma} \cdot (\hat{\mathbf{p}} \times s_E^{-1} \hat{\mathbf{p}})$, eq. 2-39 reduces to the non-relativistic Schrödinger equation for $s_E = 1$.

where the spherical symmetry of the point-like proton Coulomb potential $\hat{V} = -Z|\hat{\mathbf{r}}|^{-1}$ has been used to write the Pauli spin-orbit operator as $2Z|\hat{\mathbf{r}}|^{-3}\hat{\mathbf{S}} \cdot \hat{\mathbf{L}} = \hat{\boldsymbol{\sigma}} \cdot (\nabla \hat{V} \times \hat{\mathbf{p}})$.

Eq. 2-40 is a large component-only approximation to the Dirac equation. As compared to the non-relativistic Schrödinger Hamiltonian $\hat{h} = \hat{T} + \hat{V}$, the Pauli-like Hamiltonian \hat{h}_P of eq. 2-41 includes a number of spin-independent and -dependent relativistic correction terms, i.e. the mass–velocity, Darwin, and spin–orbit terms, respectively. As both the mass–velocity term proportional to $|\hat{\mathbf{p}}|^4$ and the spin–orbit operator are variationally unstable, the use of \hat{h}_P is mainly restricted to perturbative calculation schemes.^o The importance of \hat{h}_P comes from its illustrative character, i.e. the formal derivation of the correction terms – particularly the Pauli spin–orbit operator – from, and as an approximation scheme to, the rigorous Dirac operator \hat{h}_D .

2.3.2 Semilocal Atomic Pseudopotential Approximations

As already pointed out briefly, the pseudopotential (PP) or effective core potential approximation^{10,27,75} is conceptually different from the decoupling approximations briefly addressed above.^p As “core approximations”,^{10,79} PP calculation schemes provide effective static (non-local, possibly spin-dependent) pseudo-

^o Eq. 2-39 is also the starting point for the variational, regular approximation schemes of van Lenthe, Baerends, and Snijders,^{69,70} but s_E^{-1} is expanded in terms of $E(\hat{V} - 2m_e c_0^2)^{-1}$. The Cowan–Griffin⁷³ and Wood–Boring⁷⁴ approximations depart from the same point, but use the exact expression $s_E := 1 - (\hat{V} - E)/2m_e c_0^2$.

^p It must be noted that these decoupling approximation schemes, including the Dirac equation itself, have been discussed for 1-electron systems only, whereas PP approximations necessarily apply to N -electron systems. However, a rigorously relativistic, Lorentz-invariant N -electron theory is not known, if it exists at all.^{10,11,63} Most often, the electron–electron interaction is modeled by non-relativistic Coulomb interactions, i.e. a $\sum_{i>j} |\mathbf{r}_i - \mathbf{r}_j|^{-1}$ term in the N -electron “Dirac–Coulomb” (DC) Hamiltonian; more accurate approaches employ the approximately retarded Breit operator^{76,77}

$$-\frac{1}{2}(\hat{\boldsymbol{\alpha}}_i \cdot \hat{\boldsymbol{\alpha}}_j |\mathbf{r}_i - \mathbf{r}_j|^{-1} + ((\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j) \cdot \hat{\boldsymbol{\alpha}}_i)((\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j) \cdot \hat{\boldsymbol{\alpha}}_j)|\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j|^{-3})$$

in the N -electron DCB Hamiltonian, and may also include QED corrections.^{10,11,63,78} Therefore, it is stressed, and is understood in the following, that any N -electron Hamiltonian is necessarily approximate, and typically other approximations of, e.g., the 1-electron part enter additionally.

potentials for valence electron-only Schrödinger equations similar to eq.s 2-40 and 2-41. Relativity enters only implicitly via the PP parametrization.^{22–24}

PP theory, particularly of the energy-consistent variant, has recently been reviewed by, e.g., Dolg et al.,^{27,75,80} who also provide comprehensive bibliographies. Therefore, the following discussion does not attempt to be complete but, instead, tries to capture the points most important from a conceptual and operational point of view.

The rigorous basis for PP theory – which might be dated back to Hellmann⁸¹ and Gombás⁸² – is provided by Generalized Phillips–Kleinmann^{83,84} theory. In essence,^{10,27,79} GPK theory defines an effective “Generalized Phillips–Kleinman” N_v -electron pseudopotential operator \hat{W}_{GPK} that, if added to valence-only Hamiltonians, allows to solve Schrödinger equations for valence-only pseudo-state functions variationally, and without explicit orthogonality requirements to the (thereby excluded) core subsystem.^q Clearly, GPK and, thus, PP theory imply a frozen core approximation,^{10,79,85} i.e. assume transferability of a fixed, atomic core subsystem to a large number of different, generally non-atomic scenarios.

Operationally, \hat{W}_{GPK} is approximated²⁷ as sum of N_v 1-electron, 1-center PP operators \hat{W}_{iA} , thus defining a valence-only Hamiltonian \hat{H}_v for $N_v \leq N$ valence electrons

$$\hat{H}_v - E_{\text{BO}} \hat{1} = \sum_{i=1}^{N_v} \hat{T}_i + \sum_{i=1}^{N_v} \sum_A (\hat{V}_{iA} + \hat{W}_{iA}) + \sum_{i>j}^{N_v} \hat{g}_{ij} \quad (2-42)$$

that, for the i -th of N_v electrons, provides a molecular core–valence interaction $\sum_A (\hat{V}_{iA} + \hat{W}_{iA})$ of contributions $\hat{V}_{iA} + \hat{W}_{iA}$ centered at atom A . E_{BO} is the classical Born–Oppenheimer nuclear–nuclear, i.e. core–core repulsion energy, which consequently is to be modified according to the set of PPs \hat{W}_{iA} employed.²⁷

Similar to the (non-relativistic) ansatz of Kahn and Goddard,⁸⁶ most contemporary semilocal PPs are written as, dropping the electron and atomic core indices

^q A separation in valence and core “parts” or subsystems is, of course, not possible rigorously and assumes, in one or the other way, some kind of independent-particle theory. Whereas PK theory⁸³ considers a mean-field model from the beginning and actually is a 1-electon theory, Weeks and Rice⁸⁴ define a set of orthonormal functions spanning a core subspace.

i and *A* for clarity,

$$\hat{W} := \sum_{l=0}^{\infty} \sum_{jm_j} \hat{W}'_{lj} |lm_j\rangle\langle lm_j| \quad (2-43)$$

with 2-spinor spherical harmonics $|lm_j\rangle = \sum_{m_l m_s} C(l, m_l, s, m_s; j, m_j) |lm_l\rangle \otimes |sm_s\rangle$; the projectors provide both *l*- and *j*-dependency of the radial potential operators \hat{W}'_{lj} . Practically, all \hat{W}'_{lj} for $l \geq L$ are collected in a single local term \hat{W}_{LJ} such that, from $\sum_{l=0}^{\infty} \sum_{jm_j} |lm_j\rangle\langle lm_j| = \hat{1}$, the sum in eq. 2-43 is truncated as

$$\hat{W} \approx \hat{W}_{LJ} + \sum_{l=0}^{L-1} \sum_{jm_j} \hat{W}_{lj} |lm_j\rangle\langle lm_j|, \quad (2-44)$$

with $\hat{W}_{lj} := \hat{W}'_{lj} - \hat{W}_{LJ}$, to good approximation.^{86,87} Typically, $L - 1$ is chosen as the largest orbital angular momentum quantum number *l* occupied in the core subsystem.

Relativistic Energy-Consistent Pseudopotential Parametrization

Departing from the general relativistic PP expression of eq. 2-44, the radial potential operators \hat{W}_{lj} are mostly written as⁸⁷

$$\langle \mathbf{r} | \hat{W}_{lj} | \mathbf{r}' \rangle = W_{ljkA}(\mathbf{r}) = \sum_k C_{ljkA} |\mathbf{r} - \mathbf{r}_A|^n_{ljkA} e^{-\beta_{ljkA} |\mathbf{r} - \mathbf{r}_A|^2}, \quad (2-45)$$

where the atomic core label *A* has been explicitly included. For a given atom type and PP atomic core size, i.e. a given number $N_A < Z_A$ core electrons included in the core subspace, the parameter set $\mathcal{A}_A(N_A) := \{C_{ljkA}, n_{ljkA}, \beta_{ljkA}\}_{ljk}$, is defined according to one of a number of PP parametrization schemes.^{27,80}

In the particular relativistic energy-consistent *ab initio* PP case, $\mathcal{A}_A(N_A)$ is defined by least-square fitting procedures to valence total energies from accurate reference calculations, typically considering all *LS* or *J* levels of a large number of configurations of the atom and a number of its ions.^r Modern parametrization

^r Other approaches differ in the actual PP operator expression, the fitting procedure and target data sets considered, or both. For example, shape-consistent PPs^{10,27,79} depart also from eq. 2-44, but aim at accurate modelling of all-electron orbital or 2-spinor radial distributions (outside a cutoff radius) and energies.

schemes rely on numerical finite-difference, i.e. formally complete basis set 4-component MCDHF calculations with DC, DCB, or DCB+QED Hamiltonians and finite nucleus models^p as discussed in much detail by Cao and Dolg.²⁷

Energy-consistent PPs allow for considerable computational savings: This is not only because the excluded (possibly large) core subsystem has not to be considered in the basis set expansion. Instead, the relaxed valence–core orthogonality requirements allow pseudo-orbital or -2-spinor transformations to smooth radial distributions of considerably simplified node structures in the core region, which allow more efficient basis set expansions also of valence orbitals or 2-spinors.²⁷

Separation of Spin-Free and Spin–Orbit PP Parts

Eq. 2-44 can be,^{25,26} and frequently is, re-arranged to allow separation of a spin-free and a spin–orbit part \hat{A} and \hat{B} , respectively. Whereas this separation does not only ease the interpretation of the compact expression 2-44 – which allows for spin symmetry breaking implicitly by its dependence on the total angular momentum quantum number $j = l \pm s$ – it is also important for the generation of spin-free, i.e. scalar-relativistic PPs from, e.g., 4-component reference calculations, as the spin–orbit part \hat{B} can simply be omitted.

Following the more detailed elaboration given by Dyall and Fægri,¹⁰ writing $\hat{W} - \hat{W}_{LJ} = \sum_{l=0}^{L-1} \hat{W}_l$ in terms of contributions of a single orbital angular momentum

$$\hat{W}_l = \sum_{jm_j} \hat{W}_{lj} |ljm_j\rangle\langle ljm_j|,$$

one can define the respective contributions \hat{A}_l and \hat{B}_l to the spin-free and spin–orbit part \hat{A} and \hat{B} as

$$\hat{A}_l := \frac{l\hat{W}_{ll-} + (l+1)\hat{W}_{ll+}}{2l+1}, \quad \hat{B}_l := 2\frac{\hat{W}_{ll+} - \hat{W}_{ll-}}{2l+1}, \quad (2-46)$$

with the short-hand notation

$$l^\pm := l \pm \frac{1}{2}, \quad (2-47)$$

i.e. with l^+ for $j = l + \frac{1}{2}$ and l^- for $j = l - \frac{1}{2}$. From eq.s 2-46, the spin-free operators \hat{A}_l can be interpreted as a degeneracy-weighted average of the operators

$\hat{W}_{ll\pm}$ for the spin-orbit-split levels with quantum numbers l and $j = l \pm s$, while the spin-orbit part's \hat{B}_l operators correspond to the difference potentials.

Then, expanding the 2-spinor spherical harmonics $|ljm_j\rangle$ in terms of products of spherical harmonics and spin functions, re-arranging eq.s 2-46 for \hat{W}_{ll+} and \hat{W}_{ll-} , substituting, and collecting terms gives^{10,25,26}

$$\begin{aligned} \hat{W} &= \underbrace{\hat{W}_{LJ} + \sum_{l=0}^{L-1} \sum_{m_l} \hat{A}_l |lm_l\rangle \langle lm_l|}_{\hat{A}} + \underbrace{\hat{S} \cdot \sum_{l=0}^{L-1} \sum_{m_l m'_l} \hat{B}_l |lm_l\rangle \langle lm_l| \hat{\mathbf{L}} |lm'_l\rangle \langle lm'_l|}_{\hat{B}} \\ &= \hat{A} + \hat{B} \end{aligned} \quad (2-48)$$

where a spin space unit operator has been omitted in the spin-free part \hat{A} , that has been defined to include also the local term \hat{W}_{LJ} for later convenience. Note that the spin-orbit PP part \hat{B} is variationally stable, as compared to, e.g., the Pauli spin-orbit operator of eq. 2-41.

From eq.s 2-45 and 2-46, both \hat{A} and \hat{B} have essentially analog radial expansions, i.e., for $X_l \in \{A_l, B_l\}$,

$$X_{lA}(\mathbf{r}) = \sum_{jk} X_{ljkA} |\mathbf{r} - \mathbf{r}_A|^n_{ljkA} e^{-\beta_{ljkA} |\mathbf{r} - \mathbf{r}_A|^2} \quad (2-49)$$

for

$$A_{ll^\pm kA} =: \frac{\nu_{l^\pm} C_{ll^\pm kA}}{2l+1}, \quad B_{ll^\pm kA} =: \pm \frac{C_{ll^\pm kA}}{2l+1} \quad (2-50)$$

with $\nu_{l^\pm} := l^\pm + \frac{1}{2}$.

3

2-Component HF SCF Theory

Hartree–Fock self-consistent field (HF SCF) electronic structure theory^a is pivotal for chemistry and quantum chemistry from a number of important points. First, it provides a rigorously defined “orbital” concept⁹³ that, although – or maybe because – not employed that rigorously in chemistry,^{94–97} is central to contemporary chemists’ ideas of electronic structure and chemical bonding.^{93,98,99} Second, and more important from the point of view of computational quantum chemistry, it provides a reference for almost all wave function-based correlated methods^{100,101} and, to some extent, the basis for Kohn–Sham density functional theory.^{102–104}

The following detailed discussion of 2-component HF SCF theory – both abstract^{29,30} and in finite basis set representation²⁸ in sec.s 3.1 and 3.2, respectively – assumes closed-shell systems of $2N_v$ electrons and a Born–Oppenheimer, $2N_v$ -electron PP Hamiltonian \hat{H}_v as given by eq. 2-42, with

$$\hat{g}_{ij} := |\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j|^{-1}. \quad (3-1)$$

However, most considerations can be directly transferred to respective all-electron calculation schemes by replacing the core–valence PP interaction $\sum_A \hat{V}_A + \hat{W}_A$ with the electron–nuclei Coulomb interaction $\sum_A \hat{U}_A$.

^a Primary references are the 1928 and 1930 papers of Hartree^{88–90} and Fock,⁹¹ respectively, but important contributions have also been made by Slater.⁹²

3.1 Kramers-Restricted 2-Component HF SCF Theory

The presentation of sec.s 3.1.1 and 3.1.2 aims at a relatively detailed elaboration of the underlying general concepts at a 2-component level of HF SCF theory, closely following Almlöf¹⁰⁵ and Szabo and Ostlund.¹⁰⁰ Kramers' restriction,^{28–30} i.e. imposing time reversal invariance on the HF SCF $2N_v$ -electron state function, is discussed in sec. 3.1.3

3.1.1 The Slater Determinant Ansatz

With its placement as “approximate wave function / rigorous energy” class theory,¹⁰⁶ closed-shell HF SCF theory is defined^{47,100,105,107} by its ansatz to approximate the solution of a time-independent $2N_v$ -electron Schrödinger equation, given the “exact” $2N_v$ -electron Hamiltonian,^b by a Slater determinant

$$|\Psi\rangle = \sqrt{2N_v!} \hat{\mathcal{A}} \bigotimes_{i=1}^{2N_v} |\psi_i\rangle = \frac{1}{\sqrt{2N_v!}} \sum_P \text{sgn } P |\psi_{P(1)} \otimes \cdots \otimes \psi_{P(2N_v)}\rangle \quad (3-2)$$

of $2N_v$ 1-electron functions $|\psi_i\rangle$, and by a set of $2N_v$ 1-electron equations

$$\hat{f}_v |\psi_i\rangle = \epsilon_i |\psi_i\rangle \quad (3-3)$$

that define these 1-electron functions. From eq.s 2-42 and 2-44, the $|\psi_i\rangle$ are 1-electron 2-spinors

$$|\psi_i\rangle = |\psi_{i\alpha} \otimes \alpha\rangle + |\psi_{i\beta} \otimes \beta\rangle. \quad (3-4)$$

The motivation of the Slater determinant ansatz of eq. 3-2 is the decoupling of the $2N_v$ -electron state function $|\Psi\rangle$ with respect to the electrons' spatial and spin coordinates, giving rise to an ”independent electron picture”, while satisfying the Symmetrization and Pauli Exclusion principle:⁴⁷

Clearly, the simple Hartree product $|\psi_1 \cdots \psi_{2N_v}\rangle$ of $2N_v$ 1-electron 2-spinors $|\psi_i\rangle$ provides such a picture, as the joint probability density, i.e. the probability

^b It is clear from the point of view of the discussion of sec. 2.3, particularly in the PP approximation setting, this formal classification is problematic and, in fact, does not hold. However, considering the term “approximated wave function / rigorous energy” is instructive if opposed to the alternative “rigorous density / approximated energy”¹⁰⁶ framework of, e.g., Kohn–Sham DFT.^{102–104}

density for electrons i, j, \dots to be simultaneously at $\mathbf{r}_i, \mathbf{r}_j, \dots$ with spin projections $\gamma_i, \gamma_j, \dots \in \{\alpha, \beta\}$, respectively, factorizes as $\prod_i |\langle \mathbf{r}_i \gamma_i | \psi_i \rangle|^2$ to give $2N_v$ probabilistically independent distributions.

The antisymmetrization

$$\hat{\mathcal{A}} := \frac{1}{(2N_v)!} \sum_{P \in S_{2N_v}} (\text{sgn } P) \hat{P} \quad (3-5)$$

removes the exchange degeneracy, i.e. the $(2N_v)!$ -fold ambiguity of the representation of the $2N_v$ -electron state function by $|\psi_1 \dots \psi_{2N_v}\rangle$, by addressing all $(2N_v)!$ permutations $P \in S_{2N_v}$ of electron spatial and spin coordinates, and imposes the correct fermion permutation symmetry by weighting P with its parity $\text{sgn } P$.^c

Among the correct $2N_v$ -fermion permutation symmetry and – trivially, from the square-integrability of the normalized 1-electron 2-spinors – $2N_v$ -representability, the HF SCF Slater determinant of eq. 3-2 shares the properties of being variational; size-extensive; and, in the Born–Oppenheimer approximation, being a basis for the irreducible representations of the molecular double point group with the exact $2N_v$ -electron wave function.^{51,101}

^c Note that, therefore, $|\Psi\rangle$ by eq. 3-2 does not provide $2N_v$ probabilistically independent 1-electron probability densities: The joint probability density

$$|\langle \mathbf{r}_1 \gamma_1 \dots \mathbf{r}_{2N_v} \gamma_{2N_v} | \Psi \rangle|^2 = \frac{1}{(2N_v)!} \sum_{PQ} \text{sgn}(PQ) \prod_i \psi_{P(i)\gamma_i}(\mathbf{r}_i) \psi_{Q(i)\gamma_i}^*(\mathbf{r}_i)$$

clearly does not factorize. Moreover, the HF SCF pair density $\Pi_{\gamma_1 \gamma_2}(\mathbf{r}_1, \mathbf{r}_2)$, i.e. the probability to simultaneously find any two of the $2N_v$ electrons at positions $\mathbf{r}_1, \mathbf{r}_2$ with spin projections γ_1, γ_2

$$\Pi_{\gamma_1 \gamma_2}(\mathbf{r}_1, \mathbf{r}_2) = \sum_{ij} |\psi_{i\gamma_1}(\mathbf{r}_1)|^2 |\psi_{j\gamma_2}(\mathbf{r}_2)|^2 - \sum_{ij} \psi_{i\gamma_1}(\mathbf{r}_1) \psi_{j\gamma_2}(\mathbf{r}_2) \psi_{j\gamma_1}^*(\mathbf{r}_1) \psi_{i\gamma_2}^*(\mathbf{r}_2)$$

has a “Fermi hole”,¹⁰⁰ i.e. for $\gamma_1 = \gamma_2 = \gamma$ it is $\lim_{|\mathbf{r}_1 - \mathbf{r}_2| \rightarrow 0} \Pi_{\gamma\gamma}(\mathbf{r}_1, \mathbf{r}_2) = 0$ throughout space; contrasting, this limit vanishes everywhere for $\gamma_1 \neq \gamma_2$. Therefore, the HF SCF state function $|\Psi\rangle$ does correlate electrons, albeit only electrons with equal spin projections, as a direct consequence of the Symmetrization and Pauli Exclusion principle.⁴⁷ However, this correlation is of a purely quantum statistical character only, and $|\Psi\rangle$ is not correlated with respect to the electron–electron interaction; the latter is almost always referred to as “correlation” in quantum chemistry,^{100,101} such that $|\Psi\rangle$ is frequently discussed as “uncorrelated”.

3.1.2 Nature and Definition of The 1-Electron 2-Spinors

Employing the variational principle,^{92,101} the set of $2N_v$ 1-electron 2-spinors $|\psi_i\rangle$ is defined such that the total energy functional $E_{\text{SCF}} := \langle H_v \rangle_{\Psi}$ is stationary with respect to functional variations of any of the 2-spinors. As discussed below, this leads directly to the $2N_v$ Fock equations 3-3.

It is noted at this point that, therefore, the choice of a particular energy operator \hat{H}_v defines a respective HF SCF theory in terms of the nature of the 1-electron state functions $|\psi_i\rangle$.^d As already pointed out, the present discussion assumes a relativistic, i.e. spin-orbit PP Hamiltonian such that the general 1-electron state function is given by eq. 3-4. However, in the limiting $\hat{B} \rightarrow \hat{0}$ case of eq. 2-48, all $|\psi_i\rangle$ can be chosen as \hat{S}_z eigenfunctions and are typically referred to as spin-orbitals instead of 2-spinors.^{100,105,107}

For $E(\{|\psi_i\rangle\}_i)$ to be stationary with respect to variations of the set of 1-electron 2-spinors

$$\{|\psi_i\rangle\}_{i=1}^{2N_v} = \{|\psi_1\rangle, \dots, |\psi_{2N_v}\rangle\}, \quad \langle \psi_i | \psi_j \rangle \stackrel{!}{=} \delta_{ij} \quad (3-6)$$

it is necessary that

$$\delta L = \delta \left(E(\{|\psi_i\rangle\}_i) - \sum_{ij} \epsilon_{ij} (\langle \psi_i | \psi_j \rangle - \delta_{ij}) \right) \stackrel{!}{=} 0 \quad (3-7)$$

with $4N_v^2$ Lagrange multipliers $\epsilon_{ij} \in \mathbb{C}$. To obtain the Euler–Lagrange equations defining $\{|\psi_i\rangle\}_i$ one has, first, to express E in terms of $\{|\psi_i\rangle\}_i$.

This expression is readily provided by the Slater–Condon rules^{100,105} also for the 2-component setting, because the Slater–Condon rules only employ the $2N_v$ 1-electron functions' orthonormality and the 1-electron or 2-electron operator nature of the various terms of \hat{H}_v . Defining

$$\hat{h} := \hat{h}_1 := \hat{T}_1 + \sum_A (\hat{V}_{1A} + \hat{W}_{1A}), \quad \hat{g} := \hat{g}_{12}, \quad (3-8)$$

with the kinetic energy, core charge–electron interaction, and PP operators \hat{T} ,

^d Moreover, for a given \hat{H}_v , additional variational restrictions as, e.g., the Kramers' restriction of time reversal-invariance^{28–30} can or can not be applied to the set 1-electron state functions, which defines respective restricted or unrestricted HF SCF theories.^{51,107}

$\sum_A \hat{V}_{iA}$, and $\sum_A \hat{W}_{iA}$ for the i -th electron according to eq. 2-42, from the Slater-Condon rules

$$E(\{|\psi\rangle\}_i) = \sum_i \langle \psi_i | \hat{h} \psi_i \rangle + \frac{1}{2} \sum_{ij} \langle \psi_i \psi_j | \hat{g} \psi_i \psi_j - \hat{g} \psi_j \psi_i \rangle \quad (3-9)$$

such that the $2N_v$ Euler–Lagrange equations read^{100,105}

$$\hat{f}_v |\psi_i\rangle = \sum_j |\psi_j\rangle \epsilon_{ji}. \quad (3-10)$$

\hat{f}_v is the 1-electron Fock operator or Fockian

$$\hat{f}_v := \hat{h} + \hat{J}_v - \hat{K}_v = \sum_j (\hat{J}_{\psi_j} - \hat{K}_{\psi_j}) \quad (3-11)$$

defined in terms of the Coulomb and exchange operators \hat{J}_v and \hat{K}_v , respectively. The contributions \hat{J}_{ψ_j} and \hat{K}_{ψ_j} of the j -th 2-spinor $|\psi_j\rangle$ to \hat{J}_v and \hat{K}_v are integral operators defined in terms of their matrix elements in the 2-spinor basis, i.e.

$$\langle \psi_k | \hat{J}_{\psi_j} \psi_i \rangle = \langle \psi_k \psi_j | \hat{g} \psi_i \psi_j \rangle, \quad (3-12)$$

$$\langle \psi_k | \hat{K}_{\psi_j} \psi_i \rangle = \langle \psi_k \psi_j | \hat{g} \psi_j \psi_i \rangle. \quad (3-13)$$

Note that the exchange operator \hat{K}_v couples the spin components of two given 2-spinors, which is maybe unfamiliar from the point of view of non-relativistic HF SCF theory.^{100,105} Explicitly, from eq. 3-4

$$\begin{aligned} \langle \psi_k | \hat{K}_{\psi_j} \psi_i \rangle = & \langle \psi_{k\alpha} \psi_{j\alpha} | \hat{g} \psi_{j\alpha} \psi_{i\alpha} \rangle + \langle \psi_{k\alpha} \psi_{j\beta} | \hat{g} \psi_{j\alpha} \psi_{i\beta} \rangle \\ & + \langle \psi_{k\beta} \psi_{j\alpha} | \hat{g} \psi_{j\beta} \psi_{i\alpha} \rangle + \langle \psi_{k\beta} \psi_{j\beta} | \hat{g} \psi_{j\beta} \psi_{i\beta} \rangle \end{aligned} \quad (3-14)$$

introduces a coupling of $|\psi_{k\alpha}\rangle$ and $|\psi_{i\beta}\rangle$ and, consequently, $|\psi_{k\beta}\rangle$ and $|\psi_{i\alpha}\rangle$ that vanishes in the non-relativistic case if, as usually done,¹⁰⁷ the 2-spinors are chosen as \hat{S}_z eigenfunctions.

As both \hat{J}_v and \hat{K}_v are invariant under unitary transformations of the 2-spinors $|\psi_i\rangle$ among each other, eq.s 3-10 can be re-arranged to the pseudo-eigenvalue equations 3-3 by diagonalizing the matrix $\epsilon \in \mathbb{C}^{2N_v \times 2N_v}$ of Lagrange multipliers; because the Fock operator \hat{f}_v is hermitian and, thus, $\epsilon_{ji} = \langle \psi_j | \hat{f}_v \psi_i \rangle = \epsilon_{ij}^*$ from eq.s 3-10, such a transformation always exists, and leaves the Slater determinant

$|\Psi\rangle$ of eq. 3-2 invariant outside a phase factor.^{100,105} The 2-spinors satisfying eq. 3-3 are referred to as “canonical HF 2-spinors”.

Note, however, that eq.s 3-3 defines the 2-spinors only implicitly, as \hat{f}_v depends on $\{|\psi_i\rangle\}_i$ through $\hat{J}_v - \hat{K}_v$. Therefore, eq.s 3-3 must be solved iteratively to self-consistency.^{100,105,107e}

3.1.3 Kramers’ Restriction and Time Reversal Symmetry

It is clearly desirable to incorporate as much properties of the exact solution of the time-independent $2N_v$ -electron Schrödinger equation in any approximation to it. In fact, it is the consideration of the correct $2N_v$ -fermion permutation symmetry that gives way to HF SCF theory as discussed here.

As discussed in some detail in sec. 2.1.2, time reversal invariance is another fundamental symmetry of the electromagnetic Hamiltonian \hat{H}_v in the absence of external magnetic fields. As $|\Psi\rangle$ of eq. 3-2 is even under the transformation $\hat{\mathcal{K}}$ of eq. 2-23, i.e. $\hat{\mathcal{K}}^2 = \hat{1}$ from eq. 2-24, this time reversal invariance can be imposed on the HF SCF Slater determinant by means of “Kramers’ restriction” as discussed in the following.

Two Statements On Kramers-Restricted Slater Determinants

Before considering Kramers’ restriction of the set of 1-electron 2-spinors in detail, two statements on Slater determinants and the operation of time reversal – which have been given in a similarly by Lax⁵¹ – will be briefly addressed to outline the concept.

- A Slater determinant $|\Psi\rangle = \det(|\psi_i\rangle)_i$ is “Kramers restricted” if, for a phase ι ,

$$|\hat{\mathcal{K}}\Psi\rangle = \iota|\Psi\rangle \quad (3-15)$$

with $\hat{\mathcal{K}} = \hat{k}_1 \otimes \cdots \otimes \hat{k}_{2N_v}$; for eq. 3-15 to hold it is necessary and sufficient that $\mathbb{E}^1 := \text{span}(|\psi_i\rangle)_i$ is invariant under \hat{k}_1 .

^e It is noted in passing that questions concerning the existence and formal properties of solutions to eq.s 3-3 are far from trivial. However, almost all works in this field address non-relativistic all-electron HF SCF theory;^{106–110} in this setting, e.g., Lions¹⁰⁹ proved the existence of solutions of eq. 3-3 for $\sum_A Z_A \geq N$, if N is the number of electrons.

- If \mathbb{E}^1 is invariant under $\hat{k}_1 = \hat{k}$, the $2N_v$ 2-spinors are eigenfunctions of a \hat{k} -invariant Fock operator \hat{f}_v , i.e. $\hat{k}\hat{f}_v\hat{k}^\dagger = \hat{f}_v$.⁵¹

The first statement is clear from the anti-unitarity of $\hat{\mathcal{K}}$: If \mathbb{E}^1 is invariant under \hat{k} , i.e. if $|\psi'\rangle \in \mathbb{E}^1 \Rightarrow |\hat{k}\psi'\rangle \in \mathbb{E}^1$, then \hat{k} effects an anti-unitary transformation of any orthonormal basis $(|\psi'_i\rangle)_i$ that leaves the Slater determinant of these 2-spinors invariant outside a phase ι .^{51,100,105}

More explicitly, let $|\Psi\rangle = \det(|\psi_i\rangle)_i$, and let $\mathbf{C} \in \mathbb{C}^{2N_v \times 2N_v}$ be the matrix of coefficients of $(|\psi_i\rangle)_i$ with respect to any orthonormal basis $(|\psi'_i\rangle)_i$ of \mathbb{E}^1 , i.e. $|\psi_i\rangle = \sum_j C_{ji} |\psi'_j\rangle$. Then, time reversal effects the transformation

$$\mathbf{C} \mapsto \mathbf{C} \mathbf{U}_\iota = \mathbf{D}, \quad (3-16)$$

$$\hat{\mathcal{K}}: |\Psi\rangle = \det(|\psi_i\rangle)_i \mapsto \det \mathbf{U}_\iota \cdot \det(|\psi_i\rangle)_i =: \iota \det(|\psi_i\rangle)_i \quad (3-17)$$

with $|\iota|^2 = 1$ because \mathbf{U}_ι is unitary. Note that Hafner²⁹ proves a theorem in the opposite direction.

The second statement follows from the shell theorem:⁵¹ If \hat{k}_1 maps any $\psi_j \in \mathbb{E}^1$ to any $|\hat{k}\psi_j\rangle = |\bar{\psi}_j\rangle = \sum_l U_{jl} |\psi_l\rangle \in \mathbb{E}^1$, the j -th 2-spinor's contribution to the Coulomb operator \hat{J} transforms as

$$\langle \psi_k | \hat{J}_{\bar{\psi}_j} \psi_i \rangle = \sum_{ml} U_{imk}^* U_{ilj} \langle \psi_m \psi_j | \hat{g} \psi_i \psi_l \rangle = \langle \psi_k \psi_j | \hat{g} \psi_i \psi_j \rangle \quad (3-18)$$

because $\sum_{ml} U_{imk}^* U_{ilj} = \delta_{kj}$, and since \hat{g} is time reversal-invariant. The exchange operator \hat{K} transforms accordingly, such that

$$\hat{k} \hat{f}_v \hat{k}^\dagger = \hat{f}_v. \quad (3-19)$$

Note that, if $|\Psi\rangle$ is Kramers-restricted, every 2-spinor eigenvalue ϵ_i is even-fold degenerate:^{28,29} Clearly, similar to the proof of Kramers' theorem,

$$\hat{f}_v |\psi_i\rangle = \epsilon_i |\psi_i\rangle \Rightarrow \hat{k} \hat{f}_v |\psi_i\rangle = \hat{f}_v |\hat{k}\psi_i\rangle = \epsilon_i |\hat{k}\psi_i\rangle \quad (3-20)$$

with $\langle \psi_i | \hat{k}\psi_i \rangle = 0$ from eq. 2-22.^f

^f Of course, the eigenvalues of the Slater determinant $|\Psi\rangle$ are not even-fold degenerate, because $|\Psi\rangle$ is a $2N_v$ -electron state function and cannot be orthogonal to $|\hat{\mathcal{K}}\Psi\rangle$.

Choice of A Time Reversal-Invariant 2-Spinor Basis

It is particularly convenient to choose a 2-spinor basis such that, in eq. 3-15, $\iota = 1$, i.e. $|\hat{\mathcal{K}}\Psi\rangle = |\Psi\rangle$. This basis is a basis of N_v “Kramers pairs”,

$$(|\psi'_1\rangle, \dots, |\psi'_{N_v}\rangle) = (|\psi'_1\rangle, \dots, |\psi'_{N_v}\rangle, |\bar{\psi}'_1\rangle, \dots, |\bar{\psi}'_{N_v}\rangle) =: \mathcal{D}. \quad (3-21)$$

Such a basis \mathcal{D} always exist and is, in fact, orthonormal.^g

Because of eq. 2-21, the time reversal operator \hat{k} transforms the 2-spinor basis \mathcal{D} of eq. 3-21 as

$$\hat{k} : (|\psi'_1\rangle, \dots, |\psi'_{N_v}\rangle, |\bar{\psi}'_1\rangle, \dots, |\bar{\psi}'_{N_v}\rangle) \mapsto (|\bar{\psi}'_1\rangle, \dots, |\bar{\psi}'_{N_v}\rangle, -|\psi'_1\rangle, \dots, -|\psi'_{N_v}\rangle).$$

If the Slater determinant $|\Psi\rangle$ is constructed from the basis of eq. 3-21, i.e. if $|\Psi\rangle = \det \mathcal{D}$, one has $\mathbf{C} = \mathbf{1}_{2N_v}$, and $\hat{\mathcal{K}}$ effects the unitary transformation of eq. 3-16 with

$$\mathbf{U}_\iota = \mathbf{U}_1 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \otimes \mathbf{1}_{N_v} \quad (3-22)$$

with $\mathbf{1}_{N_v}$ the $N_v \times N_v$ unit matrix. Clearly, $\det \mathbf{U}_1 = 1$, such that $|\hat{\mathcal{K}}\Psi\rangle = |\Psi\rangle$ as intended.

In the following, a 2-spinor basis will be referred to as “Kramers-restricted” if $\mathbb{E}^1 = \text{span}(|\psi_i\rangle)_i$ is \hat{k} -invariant. The particular Kramers-restricted basis \mathcal{D} defined by eq. 3-21 will be referred to as a basis of “Kramers pairs” or “time reversal-invariant”, as, then, \mathcal{D} itself is \hat{k} -invariant outside a set of $2N_v$ phase factors ι_i .^h

^g Following Rösch,¹¹¹ $|\psi_1\rangle$ and $|\bar{\psi}_1\rangle$ are orthonormal because of eq. 2-22. Then, $|\psi_2\rangle$ can be chosen normalized in the orthogonal complement $(|\psi_1\rangle, |\bar{\psi}_1\rangle)^\perp =: \mathbb{E}_1^{1\perp}$; from eq.s 2-15 and 2-17, $|\bar{\psi}_2\rangle$ is also in $\mathbb{E}_1^{1\perp}$, etc. This procedure defines the basis of eq. 3-21 in N_v steps.

^h Note, however, that also the time reversal-invariant basis \mathcal{D} of eq. 3-21 is not uniquely defined, as one is still free to choose a symplectic unitary transformation \mathbf{S} of \mathcal{D} for that $\det \mathbf{S} = 1$. Clearly, any such transformation

$$\mathbf{S} = \begin{pmatrix} \mathbf{v} & -\mathbf{w}^* \\ \mathbf{w} & \mathbf{v}^* \end{pmatrix} \in \mathbb{C}^{2N_v \times 2N_v}$$

with $\mathbf{v} = \text{diag } v_i \in \mathbb{C}^{N_v \times N_v}$, $\mathbf{w} = \text{diag } w_i \in \mathbb{C}^{N_v \times N_v}$, and $|v_i|^2 + |w_i|^2 = 1$ for all i , has unit determinant and maps a given Kramers pair $(|\psi'_i\rangle, |\bar{\psi}'_i\rangle) \subset \mathcal{D}$ of 2-spinors to another, equivalent Kramers pair

$$(v_i|\psi'_i\rangle + w_i|\bar{\psi}'_i\rangle, -w_i^*|\psi'_i\rangle + v_i^*|\bar{\psi}'_i\rangle) =: (|\psi''_i\rangle, |\bar{\psi}''_i\rangle);$$

3.2 Kramers-Restricted Roothaan–Hall Equations

Almost all practical applications of HF SCF theory involve some kind of finite basis set expansion techniques.^{100,105,112,113} This is mainly connected to the fact that eq.s 3-3 have necessarily to be solved iteratively to self-consistency: That is, for \hat{f}_v constructed from a given set of 2-spinors, eq.s 3-3 are solved to give a refined set of 2-spinors that, in turn, allows the construction of a refined \hat{f}_v , and the process is repeated until some “convergence” criterion is met. Whereas the $2N_v$ integro-differential equations 3-3 can, in principle, be solved for the $2N_v$ 1-electron state functions $|\psi_i\rangle$ numerically by, e.g., finite-difference methods,¹¹⁴ this is practically not the case for systems other than atoms and linear molecules.

The ansatz of Roothaan¹¹² and Hall¹¹³ is a discretization of the 2-spinors not on \mathbb{R}^3 , but on the 1-electron Hilbert space \mathbb{E}^1 directly, i.e. in terms of a given set $B = (|\phi_p\rangle)_{p=1}^n$ of n “basis functions” $|\phi_p\rangle$.ⁱ Then, with

$$|\psi_i\rangle \mapsto \sum_{|\phi_p\rangle \in B} |\phi_p\rangle \langle \phi_p | \psi_i \rangle = \sum_{p\gamma} C_{pi\gamma} |\phi_p\gamma\rangle, \quad (3-23)$$

one has to solve for, and refine, the expansion coefficients $C_{pi} \in \mathbb{C}$ of the projection of $|\psi_i\rangle$ only.^{100,105,112,113}

Although the derivations of the following presentation are somewhat straightforward, the explicit expressions for the Fock and density matrix representations \mathbf{F} and \mathbf{D} , respectively, will be elaborated in detail to provide the basic working equations for the following parts. The time reversal-invariance property of \mathcal{D} and, thus, \hat{f}_v introduces a special matrix block symmetry discussed in sec. 3.2.2. Finally, sec.s 3.2.3 and 3.2.4 comment on the special $\hat{B} \rightarrow 0$ case and the nature of the basis functions typically employed, respectively.

thus, \mathcal{D} is defined only outside such a symplectic unitary transformation. As discussed in detail in sec. 7.2, Hafner and Schwarz²⁸ choose \mathbf{S} to maximize the 2-spinor basis’ “similarity” with a basis of corresponding non-relativistic spin-orbitals.

ⁱ Note that the discussion is restricted to expansion in terms of “scalar”, i.e. spin-free basis functions $|\phi_p\rangle$. 2-spinor expansion techniques, as common in the 4-component Dirac–Hartree–Fock setting, are discussed in some detail by, e.g., Dyall and Fægri¹⁰ and Reiher and Wolf.¹¹

3.2.1 Expansion in Terms of Scalar Basis Functions

The phrase “expansion in terms of scalar basis functions” is, from the point of the 2-spinor nature of the 1-electron state functions $|\psi_i\rangle$, an oversimplification. To be precise, the 2-spinors $|\psi_i\rangle$ are defined in the tensor product of the Hilbert space \mathbb{L} of square-integrable continuous $\mathbb{R}^3 \rightarrow \mathbb{C}$ functions with the spin- $\frac{1}{2}$ space $\mathbb{S} = \text{span}(|\alpha\rangle, |\beta\rangle)$, symbolically $\mathbb{E}^1 = \mathbb{L} \otimes \mathbb{S}$. Therefore, if

$$\mathbb{B} := \text{span } B = \text{span}(|\phi_p\rangle)_{p=1}^n \quad (3-24)$$

is spanned by the n scalar basis functions $|\phi_p\rangle$,^j the 2-component Roothaan–Hall equations are obtained by projection to the subspace

$$\mathbb{B} \otimes \mathbb{S} = \text{span}(|\phi_p \otimes \alpha\rangle)_{p=1}^n \cup (|\phi_p \otimes \beta\rangle)_{p=1}^n, \quad (3-25)$$

that, as compared to \mathbb{B} , includes the \hat{S}_z eigenfunctions $|\alpha\rangle$ and $|\beta\rangle$ of eq. 2-34. In other words, it is both the α - and β -spin components of the 2-spinors that are expanded in terms of the same scalar basis functions $|\phi_p\rangle \in B$.^{28k}

With eq. 3-23, the projected Fock equations 3-3 read

$$\langle \phi_p \gamma | \hat{f}_v \psi_i \rangle = \sum_{q\gamma'} f_{pq\gamma\gamma'} C_{qi\gamma'} = \sum_{q\gamma'} S_{pq\gamma\gamma'} C_{qi\gamma'} \epsilon_i; \quad (3-26)$$

$$f_{pq\gamma\gamma'} := \langle \phi_p \gamma | \hat{f}_v \phi_q \gamma' \rangle, \quad S_{pq\gamma\gamma'} := \langle \phi_p \gamma | \phi_q \gamma' \rangle = \langle \phi_p | \phi_q \rangle \delta_{\gamma\gamma'}, \quad (3-27)$$

for all $2N_v$ 2-spinors $|\psi_i\rangle$ and all $2n$ functions $|\phi_q \gamma\rangle$. As can be inferred from eq. 3-23, $C_{pi\gamma} = \langle \phi_q \gamma | \psi_i \rangle = \langle \phi_q | \psi_{i\gamma} \rangle$.

Collecting all $2n \cdot 2N_v$ equations 3-26 in a single matrix equation gives the 2-component Roothaan–Hall SCF matrix equation²⁸

$$\mathbf{FC} = \mathbf{SC}\epsilon \quad (3-28)$$

^j Whereas the basis functions are typically chosen to be $\mathbb{R}^3 \rightarrow \mathbb{R}$ functions, \mathbb{B} is understood to be defined over \mathbb{C} .

^k Note that the term “basis” is used loosely in this context. Generally, B includes functions linearly dependent within numerical accuracy, and should, in a strict sense, be referred to as a generating system. However, the term “basis” for such sets is common and will also be employed here and in the following.

with the Fockian matrix representation

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{\alpha\alpha} & \mathbf{F}_{\alpha\beta} \\ \mathbf{F}_{\beta\alpha} & \mathbf{F}_{\beta\beta} \end{pmatrix} := \left(\begin{array}{ccc|cc} f_{11\alpha\alpha} & \cdots & f_{1n\alpha\alpha} & f_{11\alpha\beta} & \cdots & f_{11\alpha\beta} \\ \vdots & & \vdots & \vdots & & \vdots \\ f_{11\alpha\alpha} & \cdots & f_{nn\alpha\alpha} & f_{n1\alpha\beta} & \cdots & f_{n1\alpha\beta} \\ \hline f_{11\beta\alpha} & \cdots & f_{1n\beta\alpha} & f_{11\beta\beta} & \cdots & f_{11\beta\beta} \\ \vdots & & \vdots & \vdots & & \vdots \\ f_{11\beta\alpha} & \cdots & f_{nn\beta\alpha} & f_{n1\beta\beta} & \cdots & f_{n1\beta\beta} \end{array} \right) \in \mathbb{C}^{2n \times 2n}$$

As already indicated in eq. 3-27, the overlap matrix $\mathbf{S} = \mathbf{S}_{\alpha\alpha} \oplus \mathbf{S}_{\beta\beta}$ has a similar, set simpler structure from spin orthogonality.

In the general case $n > N_v$, and eq. 3-28 is typically extended to also include a number of $2n - 2N_v$ “virtual” 2-spinor vector representations and corresponding 2-spinor eigenvalues ϵ_i , such that all matrices in eq. 3-28 have the common dimension $2n \times 2n$.

3.2.2 Time Reversal Invariance and Matrix Symmetry

The particular choice of a time reversal-invariant 2-spinor basis \mathcal{D} , as defined by eq. 3-21, manifests in a special structure of the matrices of eq. 3-28:²⁸

From the definition of \mathcal{D} , the expansion coefficients $C_{p\bar{i}\gamma}$ of the N_v time-reverse 2-spinors $|\bar{\psi}_i\rangle$ are related to the coefficients $C_{pi\gamma}$ of the N_v time-forward 2-spinors by

$$C_{q\bar{i}\alpha} = \langle \phi_q \alpha | \bar{\psi}_i \rangle = \langle \phi_q \alpha | (-|\psi_{i\beta}^* \alpha\rangle + |\psi_{i\alpha}^* \beta\rangle) = -C_{qi\beta}^*, \quad (3-29)$$

$$C_{q\bar{i}\beta} = \langle \phi_q \beta | \bar{\psi}_i \rangle = \langle \phi_q \beta | (-|\psi_{i\beta}^* \alpha\rangle + |\psi_{i\alpha}^* \beta\rangle) = C_{qi\alpha}^*, \quad (3-30)$$

where real-valued basis functions $|\phi_q\rangle$ have been assumed. Therefore, the HF SCF eigenvector matrix $\mathbf{C} \in \mathbb{C}^{2n \times 2n}$, seen as row vector of the $2n$ 2-spinor column vector representations $\mathbf{C}_i \in \mathbb{C}^{2n \times 1}$, i.e.

$$\mathbf{C} = (\mathbf{C}_1, \dots, \mathbf{C}_n, \mathbf{C}_{n+1}, \dots, \mathbf{C}_{2n}) = (\mathbf{C}_1, \dots, \mathbf{C}_n, \bar{\mathbf{C}}_1, \dots, \bar{\mathbf{C}}_n), \quad (3-31)$$

recovers the structure of the 2-spinor basis \mathcal{D} of eq. 3-21. Analog to the notation for the time-reverse 2-spinor $|\bar{\psi}_i\rangle := |\hat{k}\psi_i\rangle$, the time-reverse vector representation is written

$$\bar{\mathbf{C}}_i := (\mathbf{j} \otimes \mathbf{1}_n) \mathbf{C}_i^*, \quad (3-32)$$

where the complex conjugation operator in the basis B has been evaluated directly to give the complex-conjugated vector \mathbf{C}_i^* ; the symplectic form $\mathbf{j} := -2i\hat{S}_y$ is the representation of $-2i\hat{S}_y$ in the basis B .

In fact, in much the same way as \hat{k} -invariance of \mathcal{D} imposes a special structure on \mathbf{C} , \hat{k} -invariance of \hat{f}_v gives rise to very similar symmetry properties of $\mathbf{F} = \mathbf{h} + \mathbf{V} + \mathbf{W} + \mathbf{J} - \mathbf{K}$. From the point of view of sec. 3.1.3 this follows directly from eq. 3-19, as the projection of $\hat{k}\hat{f}_v\hat{k}^\dagger = \hat{f}_v$ to $\mathbb{B} \otimes \mathbb{S}$ reads¹

$$\begin{pmatrix} \mathbf{0}_n & -\mathbf{1}_n \\ \mathbf{1}_n & \mathbf{0}_n \end{pmatrix} \begin{pmatrix} \mathbf{F}_{\alpha\alpha}^* & \mathbf{F}_{\alpha\beta}^* \\ \mathbf{F}_{\beta\alpha}^* & \mathbf{F}_{\beta\beta}^* \end{pmatrix} \begin{pmatrix} \mathbf{0}_n & \mathbf{1}_n \\ -\mathbf{1}_n & \mathbf{0}_n \end{pmatrix} = \begin{pmatrix} \mathbf{F}_{\alpha\alpha} & \mathbf{F}_{\alpha\beta} \\ \mathbf{F}_{\beta\alpha} & \mathbf{F}_{\beta\beta} \end{pmatrix} \quad (3-33)$$

$$\Rightarrow \mathbf{F}_{\beta\beta} = \mathbf{F}_{\alpha\alpha}^*, \quad \mathbf{F}_{\beta\alpha} = -\mathbf{F}_{\alpha\beta}^* \quad (3-34)$$

necessarily. However, it will be explicitly shown in the following that all contributions to \mathbf{F} have the special structure defined by eq.s 3-34.^m

Explicit Expressions: 2-Spinor Energy and Core Hamiltonian Matrices

From eq.s 3-20 it is evident that the matrix ϵ of 2-spinor energy eigenvalues is of the structure

$$\epsilon = \text{diag } \epsilon_i \oplus \text{diag } \epsilon_i \in \mathbb{R}^{2n \times 2n}, \quad (3-35)$$

clearly satisfying eq.s 3-34.

For real-valued basis functions $|\phi_p\rangle$, matrix representations of spin-independent operators \hat{O} , i.e. \hat{T} , \hat{V} , and \hat{A} of eq. 2-48, are also real-valued. As, moreover, all such matrix representations decouple as $\mathbf{O} = \mathbf{O}_{\alpha\alpha} \oplus \mathbf{O}_{\beta\beta}$ from spin orthogonality, eq.s 3-34 are clearly satisfied.

¹ Note that the time reversal operator effects complex conjugation of operator matrix representations as, e.g., \mathbf{F} , by conjugation, but by (left) multiplication for 2-spinor vector representations.

^mIn their 1979 paper, Hafner and Schwarz²⁸ motivate Kramers' restriction from the opposite point of view, starting on the matrix algebra level directly: The spin-orbit PP operator matrix representation \mathbf{B} and, thus, the 1-electron part $\mathbf{h} + \mathbf{V} + \mathbf{A} + \mathbf{B}$ of \mathbf{F} , is naturally of the structure defined by eq.s 3-34 (which is connected to the special algebraic properties of the Pauli sigma operators $\hat{\sigma}_x$, $\hat{\sigma}_y$, $\hat{\sigma}_z$). Then, choosing a time reversal-invariant 2-spinor basis imposes the same structure on the \mathbf{K} matrix and, thus, on \mathbf{F} .

Considering the spin-orbit PP operator \hat{B} , evaluating the $\hat{\mathbf{L}} \cdot \hat{\mathbf{S}}$ dot product and performing spin integration gives

$$\langle \phi_p \alpha | \hat{B} \phi_q \alpha \rangle = \frac{1}{2} \sum_{lm_l} \langle \phi_q | \hat{B}_l \hat{L}_z | lm_l \rangle \langle lm_l | \phi_q \rangle = B_{pq\beta\beta}^*, \quad (3-36)$$

$$\langle \phi_p \alpha | \hat{B} \phi_q \beta \rangle = \frac{1}{2} \sum_{lm_l} \langle \phi_q | \hat{B}_l (\hat{L}_x - i\hat{L}_y) | lm_l \rangle \langle lm_l | \phi_q \rangle = -B_{pq\beta\alpha}^*, \quad (3-37)$$

where the truncation of the sum over all l , introduced in eq. 2-44, has been omitted for clarity. The second equalities, i.e. the relations $B_{pq\alpha\alpha} = B_{pq\beta\beta}^*$ and $B_{pq\alpha\beta} = -B_{pq\beta\alpha}^*$ corresponding to eq.s 3-34, follow from the fact that the $\hat{B}_l \hat{L}_r$ matrix elements between $|\phi_q\rangle$ and the spherical harmonics $|lm_l\rangle$ are purely imaginary.

Therefore, the matrix representation $\mathbf{h} = \mathbf{T} + \mathbf{V} + \mathbf{A} + \mathbf{B}$ is also time reversal-invariant – although this fact is not connected to the choice of a 2-spinor basis of Kramers pairs, as eq.s 3-36 and 3-37 are essentially independent of eq. 3-31.

Explicit Expressions: Density, Coulomb, and Exchange Matrices

Both the Coulomb and exchange operator matrix representations \mathbf{J} and \mathbf{K} are conveniently expressed in terms of the density matrix \mathbf{D} . With the Kramers-restricted HF SCF density operator

$$\hat{D} = |\Psi\rangle\langle\Psi| = \sum_{i=1}^{N_v} (|\psi_i\rangle\langle\psi_i| + |\bar{\psi}_i\rangle\langle\bar{\psi}_i|), \quad (3-38)$$

from eq.s 3-23, 3-29, and 3-30, the density matrix elements $D_{pq\gamma\gamma'}$ are

$$D_{rs\alpha\alpha} := \sum_i (C_{ri\alpha} C_{si\alpha}^* + C_{ri\beta} C_{si\beta}^*) = D_{rs\beta\beta}^*, \quad (3-39)$$

$$D_{rs\alpha\beta} := \sum_i (C_{ri\alpha} C_{si\beta}^* - C_{ri\beta} C_{si\alpha}^*) = -D_{rs\beta\alpha}^*. \quad (3-40)$$

In terms of the $D_{pq\gamma\gamma'}$, the contributions \hat{J}_{ψ_i} and $\hat{J}_{\bar{\psi}_i}$ of the i -th time-forward

and time-reverse 2-spinors $|\psi_i\rangle$ and $|\bar{\psi}_i\rangle$, respectively, are

$$\begin{aligned}\langle\phi_p\gamma|\hat{J}_{\psi_i}\phi_q\gamma'\rangle &= \langle\phi_p\psi_{i\alpha}|\hat{g}|\phi_q\psi_{i\alpha}\rangle\delta_{\gamma\gamma'} + \langle\phi_p\psi_{i\beta}|\hat{g}|\phi_q\psi_{i\beta}\rangle\delta_{\gamma\gamma'}, \\ \langle\phi_p\gamma|\hat{J}_{\bar{\psi}_i}\phi_q\gamma'\rangle &= \langle\phi_p\psi_{i\beta}^*|\hat{g}|\phi_q\psi_{i\beta}^*\rangle\delta_{\gamma\gamma'} + \langle\phi_p\psi_{i\alpha}^*|\hat{g}|\phi_q\psi_{i\alpha}^*\rangle\delta_{\gamma\gamma'},\end{aligned}$$

where spin integration has been carried out to show that \hat{J}_{ψ_i} and $\hat{J}_{\bar{\psi}_i}$ do not couple the 2-spinors' α - and β -spin components. Consequently, with eq.s 3-29, 3-30, 3-39, 3-40, and $D_{sryy}^* = D_{sr\gamma\gamma}$, the full Coulomb operator's matrix elements are

$$J_{pq\gamma\gamma'} = \sum_i \langle\phi_p\gamma|(\hat{J}_{\psi_i} + \hat{J}_{\bar{\psi}_i})|\phi_q\gamma'\rangle = 2 \sum_{rs} \text{Re } D_{sryy} g_{prqs} \delta_{\gamma\gamma'}, \quad (3-41)$$

such that the \mathbf{J} matrix decouples as the spin-independent core Hamiltonian part, i.e. as $\mathbf{J} = \mathbf{J}_{\alpha\alpha} \oplus \mathbf{J}_{\beta\beta}$, clearly satisfying eq.s 3-34.

By essentially the same reasoning one finds for the contributions \hat{K}_{ψ_i} and $\hat{K}_{\bar{\psi}_i}$ to the exchange operator \hat{K}

$$\begin{aligned}\langle\phi_p\gamma|\hat{K}_{\psi_i}\phi_q\gamma'\rangle &= \langle\phi_p\gamma\psi_{i\alpha}\alpha|\hat{g}|\psi_{i\alpha}\alpha\phi_q\gamma'\rangle + \langle\phi_p\gamma\psi_{i\alpha}\alpha|\hat{g}|\psi_{i\beta}\beta\phi_q\gamma'\rangle \\ &\quad + \langle\phi_p\gamma\psi_{i\beta}\beta|\hat{g}|\psi_{i\alpha}\alpha\phi_q\gamma'\rangle + \langle\phi_p\gamma\psi_{i\beta}\beta|\hat{g}|\psi_{i\beta}\beta\phi_q\gamma'\rangle, \\ \langle\phi_p\gamma|\hat{K}_{\bar{\psi}_i}\phi_q\gamma'\rangle &= \langle\phi_p\gamma\psi_{i\beta}^*\alpha|\hat{g}|\psi_{i\beta}^*\alpha\phi_q\gamma'\rangle - \langle\phi_p\gamma\psi_{i\beta}^*\alpha|\hat{g}|\psi_{i\alpha}^*\beta\phi_q\gamma'\rangle \\ &\quad - \langle\phi_p\gamma\psi_{i\alpha}^*\beta|\hat{g}|\psi_{i\beta}^*\alpha\phi_q\gamma'\rangle + \langle\phi_p\gamma\psi_{i\alpha}^*\beta|\hat{g}|\psi_{i\alpha}^*\beta\phi_q\gamma'\rangle,\end{aligned}$$

such that

$$K_{pq\gamma\gamma'} = \sum_i \langle\phi_p\gamma|(\hat{K}_{\psi_i} + \hat{K}_{\bar{\psi}_i})|\phi_q\gamma'\rangle = \sum_{rs} D_{sryy'} g_{prsq}; \quad (3-42)$$

$$K_{pq\alpha\alpha} = K_{pq\beta\beta}^*, \quad K_{pq\alpha\beta} = -K_{pq\beta\alpha}^*. \quad (3-43)$$

Eq.s 3-43 follow directly from the time reversal invariance of \mathbf{D} , as the 4-index integrals g_{prsq} are real for real-valued basis functions $|\phi_p\rangle$. As already pointed out in sec. 3.1.2, \mathbf{K} , as opposed to \mathbf{J} , does break spin symmetry, i.e. couples the 2-spinors' α - and β -spin components through the generally non-vanishing $\mathbf{K}_{\alpha\beta}$ and $\mathbf{K}_{\beta\alpha}$ blocks.

3.2.3 Spin Component Decoupling in the $\hat{B} \rightarrow \hat{0}$ Limit

For vanishing spin-orbit PP operators \hat{B} , i.e. in a scalar- or non-relativistic setting with $\mathbf{W} = \mathbf{A}$ or $\mathbf{W} = \mathbf{0}_{2n}$, respectively, the complete core Hamiltonian matrix representation \mathbf{h} decouples with respect to the spin indices. Moreover, from eq.s 3-34, $\mathbf{h}_{\alpha\alpha} = \mathbf{h}_{\beta\beta} \in \mathbb{R}^{n \times n}$. Then, all $2n$ HF SCF eigenvectors \mathbf{C}_i can be chosen real-valued and to represent \hat{S}_z eigenfunctions, i.e. to satisfy $\mathbf{S}_z \mathbf{C}_i = \pm \frac{1}{2} \mathbf{C}_i$, provided that the first $\mathbf{J} - \mathbf{K}$ matrix in the SCF loop is constructed from an initial guess that does not break spin symmetry.

Then, the density matrix decouples accordingly because, in eq. 3-40, the sum runs over differences of products that always include at least one vanishing factor as the i -th eigenvector \mathbf{C}_i has either $C_{pi\alpha} = 0$ or $C_{pi\beta} = 0$ for all p . Thus, from eq. 3-42 also $\mathbf{K}_{\gamma\gamma'} = \mathbf{0}_n$ for $\gamma \neq \gamma'$, and the 2-component Roothaan–Hall equation 3-28 decouples to give two real-valued $n \times n$ equations

$$\mathbf{F}_{\alpha\alpha} \mathbf{C}_\alpha = \mathbf{S}_{\alpha\alpha} \mathbf{C}_\alpha \epsilon \quad \text{and} \quad \mathbf{F}_{\beta\beta} \mathbf{C}_\beta = \mathbf{S}_{\beta\beta} \mathbf{C}_\beta \epsilon. \quad (3-44)$$

Clearly, both are identical from the imposed Kramers' restriction – which is, in fact, identical with the spin restriction of non-relativistic RHF SCF theory^{100,105} in this case –, and one has the simplified relations

$$D_{pq} = 2 \sum_i C_{pi} C_{qi}, \quad J_{pq} - K_{pq} = \sum_{rs} D_{sr} (2g_{prqs} - g_{prsq}), \quad (3-45)$$

where the spin indices have been dropped.

3.2.4 Cartesian Gaussian-Type Orbital Basis Functions

So far, outside the assumption $\phi_p(\mathbf{r}) \in \mathbb{R}$ for all \mathbf{r} , nothing has been stated about the nature of the functions $|\phi_p\rangle \in B$ used to span the subspace $\mathbb{B} \otimes \mathbb{S}$. No review of the wealth of functions that can possibly be employed is given here. The book of Helgaker, Jørgensen, and Olsen¹⁰¹ provides both an in-depth discussion of the subject and a large number of references.

In most wave function-based electronic structure calculations, the basis B is chosen as a set of Gaussian-Type Orbitals, GTOs, i.e. of typically atom-centered functions with Gaussian functions – or linear combinations of Gaussian functions – for the radial part.

Most often these are Cartesian GTOs, i.e. CGTOs, of the form

$$\phi_p(\mathbf{r}) = N_p \cdot (x - x_p)^{m_{px}} (y - y_p)^{m_{py}} (z - z_p)^{m_{pz}} \cdot e^{-\xi_p |\mathbf{r} - \mathbf{r}_p|^2}, \quad (3-46)$$

centered at \mathbf{r}_p . The Cartesian monomials $m_{px}, m_{py}, m_{pz} \in \mathbb{N}$ are connected to the CGTO's associated orbital angular momentum quantum number l_p via $m_{px} + m_{py} + m_{pz} =: \|\mathbf{m}_p\|_1 = l_p$, introducing the Cartesian monomial vector $\mathbf{m}_p \in \mathbb{N}^3$ and its 1-norm for later convenience.ⁿ

The main motivation of choosing CGTO basis functions for the discretization of, e.g., the Hartree–Fock SCF equations, is connected to the dramatic simplification of the evaluation of 1-electron and, to some extent, 2-electron operator matrix elements between CGTOs^{101,115} – as compared to, e.g., exponential-type functions. In fact, the convergence of the expansion of eq. 3-23 is slow with respect to increasing size of B ,¹⁰¹ particularly in the 2-component setting.¹¹⁶ However, as most matrix elements between CGTOs can be evaluated analytically or employing at most a single one-dimensional numerical integration, the relatively large size of CGTO basis sets is typically outweighed by far.

3.3 Closing Comments on Correlation

It is clear that the time-independent Schrödinger equation 2-5 is not solved by the HF SCF Slater determinant ansatz of eq. 3-2. In fact, as pointed out in sec. 3.1.1, $|\Psi\rangle$ of eq. 3-2 incorporates only the quantum statistical correlation of the $2N_v$ electrons due to their nature as indistinguishable, elementary spin- $\frac{1}{2}$ particles, but not correlation due to their physical interaction.

It is this characteristic feature of the HF SCF $2N_v$ -electron state function that gives rise to the definition of the correlation energy^{100,101}

$$E_{\text{corr.}} := \mathcal{E} - E_{\text{SCF}} \quad (3-47)$$

ⁿ The set of all CGTOs includes the spherical harmonic-GTOs from the possibility to re-write the real spherical harmonics Y_l^m in terms of only the Cartesian coordinates x, y, z , such that every spherical harmonic-GTO with a given orbital angular momentum quantum number l is a linear combination of CGTOs with $\|\mathbf{m}\|_1 = l$.¹⁰¹ Note, however, that for a given l there are $\frac{1}{2}(l+1)(l+2)$ CGTOs, but only $2l+1$ spherical harmonic-GTOs.

Closing Comments on Correlation

in terms of the “exact” energy \mathcal{E} and the expectation value $E_{\text{SCF}} = \langle H_v \rangle_\Psi$ of \hat{H}_v with the Slater determinant $|\Psi\rangle$.

Probably the largest part of methods developed, and still being developed, in computational quantum chemistry is concerned with the accurate, yet mostly approximate computation of $E_{\text{corr.}}$,^{100,101} and this is maybe also true for relativistic quantum chemistry.^{10,11,78} In principle, the exact solution of eq. 2-5 – still assuming a $2N_v$ -electron closed-shell system – is readily written as the (full) Configuration Interaction state function

$$|\text{full CI}\rangle \quad (3-48)$$

$$:= |\Psi\rangle + \sum_{ia} C_i^a |\Psi_i^a\rangle + \frac{1}{(2!)^2} \sum_{ijab} C_{ij}^{ab} |\Psi_{ij}^{ab}\rangle + \frac{1}{(3!)^2} \sum_{ijkabc} C_{ijk}^{abc} |\Psi_{ijk}^{abc}\rangle + \dots$$

in terms of $|\Psi\rangle$ of eq. 3-2: The determinants $|\Psi_i^a\rangle$, $|\Psi_{ij}^{ab}\rangle$, $|\Psi_{ijk}^{abc}\rangle$, … with single, double, triple, … substitutions, respectively, are defined in terms of $|\Psi\rangle$ in the sense that, e.g., $|\Psi_i^a\rangle$ is obtained from $|\Psi\rangle$ by substituting the i -th 2-spinor with the a -th of the $2n - 2N_v$ “virtual” 2-spinors.^o However, for a given basis set expansion of the 2-spinors in terms of n basis functions $|\phi_p\rangle \in B$, the number $N_{\text{CI}}(m)$ of m -fold substituted determinants is¹⁰⁰

$$N_{\text{CI}}(m) = \binom{2N_v}{m} \binom{2n - 2N_v}{m}, \quad (3-49)$$

and, thus, the number of terms included in eq. 3-48 in almost all cases too large. Therefore, $E_{\text{corr.}}$ is almost always computed in terms of approximations to $|\text{CI}\rangle$.

A wealth of hierarchies of approximate correlated electronic structure methods has been and is still being developed,^{10,11,78,100,101,117} and it is both impossible and inappropriate to attempt any systematic discussion here.

A successful class of approximations to $E_{\text{corr.}}$ of eq. 3-47 is not obtained from perturbation theory:^{100,101} Partitioning the $2N_v$ -electron Hamiltonian \hat{H}_v as

$$\hat{H}_v =: \hat{F} + \lambda \hat{\mathcal{V}} \quad (3-50)$$

^o $|\text{CI}\rangle$ is, in principle, exact because the set of all $2N_v$ -electron Slater determinants is a complete orthonormal system of functions for the $2N_v$ -electron state Hilbert space. However, this assumes a complete set of 1-electron state functions $|\psi_i\rangle$.

in terms of an operator \hat{F} of which eigenfunctions and spectrum are known, and a “small” perturbation $\hat{\mathcal{V}}$, Rayleigh–Schrödinger perturbation theory^{44,100,118} expands both eigenfunctions and eigenvalues of \hat{H}_v in a Taylor series in the ordering parameter $\lambda \in [0, 1]$. Clearly, as $\lambda \rightarrow 1$, the original $2N_v$ -electron Hamiltonian \hat{H}_v is recovered from eq. 3-50.

A particular common choice of the \hat{F} and $\hat{\mathcal{V}}$ operators for the approximate calculation of the correlation energy $E_{\text{corr.}}$ is¹¹⁹

$$\hat{F} := \sum_i \hat{f}_{vi}, \quad \hat{\mathcal{V}} := \sum_{i>j} \hat{g}_{ij} - \sum_i (\hat{J}_{vi} - \hat{K}_{vi}) \quad (3-51)$$

with the valence-only Fockian \hat{f}_{vi} , Coulomb operator \hat{J}_{vi} , and exchange operator \hat{K}_{vi} for the i -th electron, defining Møller–Plesset perturbation theory.^{100,101} Clearly, the lowest-energy eigenfunction of \hat{F} is the $2N_v$ -electron Slater determinant of eq. 3-2 with eigenvalue $\sum_i \epsilon_i$. With eq. 3-51, expanding the total energy up to second order in λ gives the Møller–Plesset correlation energy to second order,^{100,101p}

$$E_{\text{MP2}} := \frac{1}{4} \sum_{ijab} \frac{|g_{ijab} - g_{ijba}|^2}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}, \quad (3-52)$$

where i and j label occupied, and a and b label virtual 2-spinors.

^p Note that, as discussed in detail by, e.g., Helgaker, Jørgensen, and Olsen,¹⁰¹ the Møller–Plesset perturbation expansion is not guaranteed to converge, and can, in fact, diverge in cases of practical interest. However, it is probably fair to state E_{MP2} is most often a good estimate of the correlation energy.

4

Scope, Strategy, Status Quo 2006

The main goal of this work is the development of a Kramers-restricted 2-component pseudopotential Hartree–Fock SCF program on the basis of the “Quantum Objects Library” HF SCF modules.

Linking the preceding theoretical with the following technical, implementation part, this chapter is intended to provide a short discussion of this work’s prerequisites, scope, and the global strategy employed to approach the research goals formulated. In fact, these considerations reflect to a large extent the structure and organization of the following chapters, which is briefly outlined and reasoned in sec. 4.2. Finally, sec. 4.3 provides a technical discussion of a small part of the 2006 QOL implementation’s matrix, integral evaluation, and HF SCF modules.

4.1 The “Quantum Objects Library”

The “Quantum Objects Library”, QOL, is a set of program modules that provides a modular, highly abstract structure mainly for correlated wave function-based *ab initio* electronic structure methods with emphasis on (arbitrary excitation single- and multi-reference) Coupled Cluster theory and variants thereof. Technically, the largest part is written in C++, including a number of code-generated components and interfaces to standard libraries as, e.g., the LAPACK/BLAS libraries. The QOL has been initiated, designed, and developed by M. Hanrath and coworkers at

Cologne University.

Around the end of 2006, the QOL also provided a small number of less developed modules for CGTO integral evaluation and integral-conventional spin-restricted HF SCF calculations. The importance of this part of the QOL does not originate in the competitive performance and functionality characteristics of the HF SCF program provided – in fact, with respect to these modules, the 2006 QOL implementation has to be considered as exploratory only. Instead, its relevance comes from the connection to the well-developed CC modules, and from its nature as in-house development that allows access to, and modification of, the source code.

4.2 Scope, Strategy, and Organization of This Work

This work builds heavily on the HF SCF (and a number of related) QOL modules and is, thus, to be understood as part of the ongoing development process at the group of M. Dolg at Cologne University.

The term “2006 QOL implementation” can, of course, only be used in a somewhat loose sense, as particularly the QOL’s CC modules have been continuously modified and extended by M. Hanrath and coworkers. However, the QOL’s HF SCF-related modules have only been modified and extended in the course of this work, such that the term “2006 QOL implementation”, if employed with respect to these modules, refers to the QOL HF SCF parts’ status quo prior to the modifications and extensions presented here, and does so from end 2006 to ca. spring 2010.

4.2.1 Implementation Goals

Starting from the 2006 QOL implementation’s 1-component spin-restricted HF SCF modules, this work aims at

- the implementation of both spin-averaged and spin-orbit pseudopotential integrals over generally contracted CGTO basis functions;
- the integration of these functionalities in the established QOL HF SCF modules, i.e. the development of both 1-component spin-restricted all-electron

- and spin-free PP, and 2-component Kramers-restricted spin-orbit PP HF SCF programs; and
- the development of programs for 4-index integral transformations to the molecular 2-spinor basis, and for subsequent 2-component Møller–Plesset perturbation theory calculations.

Additionally, much effort has been made to guarantee and, to some extent, enable applicability of the developed HF SCF modules to the use-cases considered (e.g. by improving the 2006 QOL implementation’s naïve Roothaan–Hall SCF algorithm and core Hamiltonian initial guess).

Strategically, it is clear from the introductory discussion of sec. 4.1 that *any approach of these implementation goals is to be made within the primary requirements of maintaining interoperability and compatibility with the established modules* – both technically and in terms of the object-oriented programming paradigm employed^a –, *as well as with the modules being subject of ongoing development*. It is stressed again that, from this point of view, this work must not be understood as independent, but as part of the global QOL structure as it stands around spring 2010.

4.2.2 Implementation Strategy and Outline

Because of the nature of this work as part of the ongoing QOL development process, most of this work’s implementation has necessarily been done in C++.^b

Globally, a bottom-up strategy was employed to approach the implementation goals formulated. The following, second part of this work is organized essentially analog to this strategy which is, thus, given in some detail to outline the former.

^a It is noted in passing that, from the point of view of the formulated research goals, the 2006 QOL implementation both enables and limits this work: As, for example, the development of the QOL integral evaluation modules is not, and cannot be, the scope of this work alongside with what has been stated, the HF SCF programs developed in this work are logically restricted by the 2006 QOL implementation’s exploratory status.

^b A review of object-oriented scientific programming in FORTRAN90 and C++, introducing C++ concepts as, e.g., inheritance, polymorphism, and template techniques from the point of view of particle physics, has been given by Cary et al.¹²⁰

- From the spin symmetry-breaking nature of the spin-orbit PP operator \hat{W} of eq. 2-44 one has to consider the full $\mathbb{C}^{2n \times 2n}$ Roothaan–Hall SCF equations instead of two smaller, spin symmetry-related $\mathbb{R}^{n \times n}$ problems. Consequently, in the very first step the 2006 QOL implementation’s matrix algebra modules were complexified and supplemented by matrix classes exploiting time reversal-related index symmetry. In a subsequent step these were connected to the established QOL iterator–evaluator structure, allowing the assembly of all non-PP matrices in the extended $\mathbb{C}^{2n \times 2n}$ framework. The implementation details are discussed in ch. 5.
- In a next intermediate step, PP classes and the necessary PP parameter input functionalities have been introduced, including the 2006 QOL implementation’s all-electron calculation scheme as special case. Then, with the necessary matrix algebra framework established, spin-averaged and spin-orbit PP integrals have been implemented by interfacing the PP integral subroutines^{31,32} of the ARGOS integral program^{31–35} as discussed in ch. 6.
- The third step, addressing the 2-component PPs in the HF SCF framework, was to
 - implement complex-valued hermitian eigenvalue equation solving and spin symmetry-broken density matrix assembly;
 - implement spin component exchange coupling according to eq. 3-42; and
 - organize SCF eigenvector processing to impose and maintain time reversal-invariance according to eq. 3-31 over the iterative solution of eq. 3-28.

Improvements of SCF convergence and initial guessing, i.e. the implementation of a 2-component “Optimal Damping Algorithm”^{36,37} and “Molecule-from-Atoms” density matrices,³⁸ are also discussed in ch. 7.

4-index integral transformation to the Fockian eigenbasis is, to a greater or lesser extent, independent from the HF SCF framework and is presented in ch. 8.

Generally, the presentations given in the upcoming chapters 5, 6, 7, and 8 – and, consequently, also the following discussion of parts of the 2006 QOL implementation in sec. 4.3 – have been given a strong technical focus, and have been set

up close to the source code. This necessarily goes at the expense of the reader who is not interested in implementation details. However, much of the value of the accessibility of the source code comes from its documentation. From this and the point of view of sec. 4.1, ch.s 5–8 have been written with the particular aim of providing a documentation of the design, implementation, and, to some extent, the source code, i.e. with the hope to be able to assist future development.

4.3 Quantum Objects Library: Status Quo End 2006

The following discussion of a smaller number of selected modules and class structures of the 2006 QOL implementation cannot aim at a self-contained documentation and is far from comprehensive. Instead, it presents code design features that guided and – for compatibility reasons – restricted design and implementation of new modules and class structures. Particularly the introduction of complex-valued matrix classes and the interface of the ARGOS PP integral subroutines discussed in ch.s 5 and 6, respectively, are closely interrelated to these parts of the 2006 QOL implementation.

Consequently, sec.s 4.3.1, 4.3.2, and 4.3.3 are set up – and meant to be limited – to provide a preparatory technical discussion for the upcoming ch.s 5, 6, and, to some extent, 7.

4.3.1 QOL Matrix and Matrix Representation Classes

Generally, QOL matrix class design and implementation is characterized by the strict separation from matrix entry-storing “container–evaluator”, and upper level matrix ”algebra” classes:

Every matrix object makes reference to an associated “container–evaluator” object that defines memory allocation, layout, and access via a container class holding the matrix entries, and an index operator for structured random access mimicking the possibly non-trivial matrix structure.^c The matrix ”algebra”, i.e.

^c In principle, a container–evaluator class does not need to allocate any memory at all. Matrix elements could as well be re-computed at access by connecting the index operator to, e.g., integral-processing subroutines. The particular HF SCF setting does, however, employ container–evaluator

the set of permitted operations relating matrices to others and other quantities, is common to all matrix objects independent of their internal structure, and implemented as a set of matrix class methods and operators. This gives way to the easy

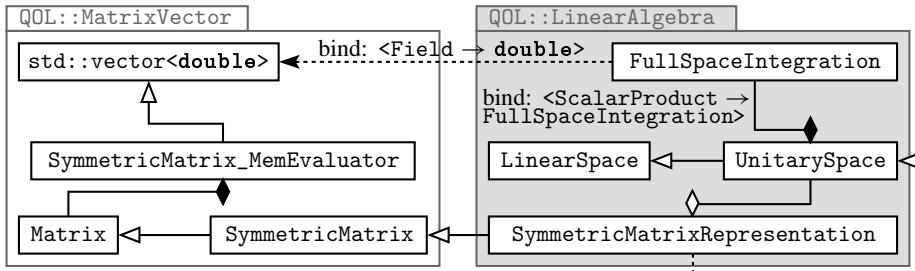


Fig. 4.1: UML class diagrams for parts of the `QOL::MatrixVector` and `QOL::LinearAlgebra` namespaces. All template parameters and most implied template argument bindings have been omitted for clarity. For example, the `SymmetricMatrix_MemEvaluator` attribute of `Matrix` implies binding of the `Evaluator` template argument of `SymmetricMatrix` to `SymmetricMatrix_MemEvaluator`.

and logical implementation of different types of matrices by inheritance from, and by letting the container–evaluator class be a template argument of, the basic `Matrix` class.

Fig. 4.1 illustrates the interdependence of the container–evaluator and matrix algebra classes for the particular `SymmetricMatrix` class case:

`SymmetricMatrix` is a three-parameter template class derived from `Matrix`. Matrix symmetry enters through the second, typically `SymmetricMatrix_MemEvaluator`-valued template argument, that stores the $\frac{1}{2}n(n+1)$ symmetry-unique matrix entries, i.e. the upper or lower triangular part of $\mathbf{M} \in \mathbb{R}^{n \times n}$, as defined by the `MatrixPackageOrder` template argument. Precisely, `SymmetricMatrix_MemEvaluator` has a `std::vector` attribute that holds the matrix entries sequentially for efficient one-index random accesss. Both the two-fold indexed structure and the generation of symmetry-redundant from symmetry-unique entries is provided by the two-index `()` operator

classes that hold all (symmetry-unique) matrix entries in memory, such that the term “container–evaluator” can be understood synonymeously with “container”.

```

1 const value_type & operator () (int p, int q) const {
2     if ( matrixPackageOrder == LowerDiagonal ) {
3         if( p < q )
4             std::swap(p, q);
5         return _v[p*(p+1)/2 + q] ;
6     } // ...

```

which is therefore necessarily an attribute of the `SymmetricMatrix_MemEvaluator` class.

Algebraic operations that define sums, products, etc. of matrices, including (but not restricted to) symmetric matrices, are defined in terms of methods and operators of the `Matrix` base class, and can be accessed by, e.g., `SymmetricMatrix` through the inheritance relationship.

Most matrices \mathbf{M} in the HF SCF setting are actually referred to as matrix representations \mathbf{O} of operators \hat{O} in finite-dimensional subspaces \mathbb{B} spanned by CGTO basis functions $|\phi_p\rangle \in B = (|\phi_p\rangle)_{p=1}^n$. The connection of \mathbf{O} and \hat{O} is established by

$$\hat{O} \equiv \sum_{|\phi_p\rangle, |\phi_q\rangle \in B} |\phi_p\rangle\langle\phi_p| \hat{O} \langle\phi_q|\phi_q\rangle = \sum_{pq} O_{pq} |\phi_p\rangle\langle\phi_q|, \quad (4-1)$$

which makes reference to a scalar or inner product $\mathbb{B} \times \mathbb{B} \rightarrow \mathbb{R}$, $(|\phi_p\rangle, |\phi_q\rangle) \mapsto \langle\phi_p|\phi_q\rangle$ for the definition of the matrix representation's matrix elements $(\mathbf{O})_{pq} = O_{pq} = \langle\phi_p|\hat{O}\phi_q\rangle$.^d

This abstract structure is mimicked through the design of the QOL matrix representation modules as illustrated in fig. 4.1 for the particular `SymmetricMatrixRepresentation` case: The top-level `SymmetricMatrixRepresentation` class is a two-parameter template class of a first `UnitarySpace`, and a particular matrix class second template argument, which is set to `SymmetricMatrix` by default. In fact, `SymmetricMatrixRepresentation` is such a `SymmetricMatrix` by inher-

^d Note that the inner product in eq. 4-1 can generally not be restricted to the domain $\mathbb{B} \times \mathbb{B}$ and, therefore, cannot be considered as a property of \mathbb{B} . This is because in eq. 4-1 the function $|\hat{O}\phi_q\rangle$ is not necessarily in \mathbb{B} ; the projection $|\phi_p\rangle\langle\phi_p| \hat{O} \langle\phi_q|\phi_q\rangle$ is, of course, but requires the definition of the inner product of $|\phi_p\rangle \in \mathbb{B}$ and $|\hat{O}\phi_q\rangle \in \mathbb{B}' \subseteq \mathbb{B}$, i.e. on some “larger” space \mathbb{B}' that contains \mathbb{B} as a proper subspace.

Whereas this point might be referred to as formal, it actually is of particular importance for the generalization of eq. 4-1 to include spin-orbit pseudopotential operators as discussed in 5.2.1.

itance, thus inheriting all matrix algebra operations shared with general (symmetric) matrices that do not necessarily represent operators. `UnitarySpace` provides the inner product information and, thus, defines the `SymmetricMatrix` classes' `value_type` template argument, i.e. the matrix entries' data type, as `UnitarySpace::ScalarProduct::_Field`.^e

Eq. 4-1 provides a natural connection of the construction of matrix representation objects and the evaluation of operator matrix elements, i.e. integrals, through the CGTO-spanned unitary subspace $\mathbb{B} = \text{span } B$. Consequently, QOL integrals are evaluated not before the construction of, e.g., `SymmetricMatrixRepresentation` from abstract `ContainerRepresentation` objects that carry the relevant function space and operator information, as briefly discussed in the next section.

4.3.2 QOL Iteration, Integration, and Integral Communication

The link between the QOL matrix algebra and matrix representation modules on the one side, and the integral-evaluating modules on the other side is effectively provided by the `StandardMolecularHamiltonianRepresentation` class:

Generally, top-level construction of matrix representation objects involves respective `StandardMolecularHamiltonianRepresentation` methods as, e.g., `overlap()`, and handing the `ContainerRepresentation` return type objects to the `SymmetricMatrixRepresentation` constructor:^f For example,

^e In a non- or scalar-relativistic spin-free HF SCF setting, operators are represented in a subspace $\mathbb{B} \subseteq \mathbb{L}$ of the Hilbert space of square-integrable continuous $\mathbb{R}^3 \rightarrow \mathbb{R}$ functions over \mathbb{R} , spanned by n CGTOs $|\phi_p\rangle$. \mathbb{B} inherits the \mathbb{L} inner product

$$(|\phi_p\rangle, |\phi_q\rangle) \mapsto \langle \phi_p | \phi_q \rangle = \int_{\mathbb{R}^3} d\mathbf{r} \phi_p^*(\mathbf{r}) \phi_q(\mathbf{r}) \in \mathbb{R}.$$

Consequently, `UnitarySpace` typically is of a `CartesianGaussian_UnitarySpace1` type, its `ScalarProduct` template argument being set to `FullSpaceIntegration` by default. Then, the `SymmetricMatrix` template argument `value_type` is set to the same type as the CGTO's domain data type by the `typedef typename Vector::_Field _Field` type definition in `FullSpaceIntegration`, i.e. a single or double precision floating point representation of \mathbb{R} .

^f Whereas there is no “operator” that corresponds to the overlap matrix S , i.e. the subspace \mathbb{B} 's metric in the B representation, S is technically represented by a `SymmetricMatrixRepresentation` object from the close connection to eq. 4-1 with $\hat{O} = \hat{1}$.

```

1     typedef StandardMolecularHamiltonianRepresentation<
2         CGTOBasis> H;
3 H h( molecule, basis );
4
5         // OVERLAP MATRIX CONSTRUCTION USE CASE
6 SymmetricMatrixRepresentation<H::USpace1>
7     overlapMatrix( h.overlap() );

```

`ContainerRepresentation` is derived from the `GContainer` class. The `GContainer`'s four template parameters control, in an abstract way,

- the integral, i.e. the CGTO-spanned subspace's inner product codomain via `UnitarySpace::ScalarProduct::Field`;
- this subspace, to be understood as “index” space of a set B of CGTOs, from which a set $I \subset B \times B$ of CGTO pairs can be constructed to iterate over;^g
- this iteration scheme over these CGTO pairs $\{|\phi_p\rangle, |\phi_q\rangle\}_{pq} \in I$,^g and a rule to evaluate the corresponding matrix element integral O_{pq} , via the `Buffered_IteratorEvaluator1`-valued template argument `Iterator`; and
- memory allocation, layout, and access via the `Evaluator` template argument.

Stated briefly – and discussed in more technical detail below –, it is this massively templated structure that allows the simple, intuitive handling of matrix representation class objects, showcased in lines 6–7 of the `S` matrix use-case, through the underlying QOL iterator–evaluator hierarchy. This iterator–evaluator hierarchy is defined by `GContainer`'s third template argument `Iterator` – which has, in the general 1-electron operator case, the value `Buffered_IteratorEvaluator1` –, and provides all information for the construction of matrix representation objects as, e.g., `SymmetricMatrixRepresentation`.

^g The set notation “ $I \subset B \times B$ ” is somewhat of an understatement from the complicated internal structure of the CGTO basis functions: As briefly discussed in sec. 3.2.4, a given CGTO $|\phi_p\rangle$ is defined by \mathbf{r}_p, l_p, ξ_p , and \mathbf{m}_p , and any iteration scheme over a set of pairs of CGTO is an 8-loop over quantities $(\mathbf{r}_p, l_p, \xi_p, \mathbf{m}_p; \mathbf{r}_q, l_q, \xi_q, \mathbf{m}_q)$. A particular loop nesting can be significantly more efficient from the point of view of, e.g., integral evaluation and pre-screening.¹²¹ Thus, I should more precisely be referred to as a sequence of 8-tupels generated from a subset of $B \times B$, instead of only an (unstructured) subset of $B \times B$.

Coarse-Grain Overview of the QOL Iterator–Evaluator Hierarchies

The two most central concepts of the 2006 QOL implementation’s iterator–evaluator hierarchy are:

- a top-level black box class structure applying to all 1-electron and 2-electron operator matrix elements, i.e. 2-index and 4-index integrals, that wraps possibly complicated integral-batched iteration schemes over (hermitian) symmetry-unique integral index combinations in a common, STL-like interface; and
- a rigorous separation of iteration and the objects iterated over, i.e. of iterator and container classes.

Both design elements relate directly to the `Buffered_IteratorEvaluator` class. The general `Buffered_IteratorEvaluator` is a two-parameter template class of `Iterator` and `Evaluator` arguments, both being an `Iterator` by inheritance, and having `Iterator-` and `Evaluator-type` attributes.

The 1-electron operator case class `Buffered_IteratorEvaluator1` is derived by setting the `Iterator` argument to `CGBTree_HermitianTupel2_Iterator`, and by defining the nature of the integrals $O_{pq} = \langle \phi_p | \hat{O} \phi_q \rangle$ to be evaluated, i.e. the 1-electron operator \hat{O} referred to, by the `Evaluator` argument.

The base class `Buffered_IteratorEvaluator` provides attributes, methods, and operators necessary for an STL-like iterator interface, i.e. `valid()` and `reset()` methods, and increment `++` and de-reference `*` operators. Integral evaluation by `readAhead()` is directly coupled to the `++` operator:^h

```
1 virtual int readAhead() = 0;
2
3 Buffered_IteratorEvaluator & operator ++ () {
4     Iterator:: operator ++ ();
5     if ( ++_i >= _n ) {
6         _i = 0;
7         _n = readAhead();
8     }
9 }
```

^h `readAhead()` is `virtual` to allow derived class-specific, i.e. a 1-electron operator case-specific integral batching as implemented for the `Buffered_Iterator-Evaluator1` class.

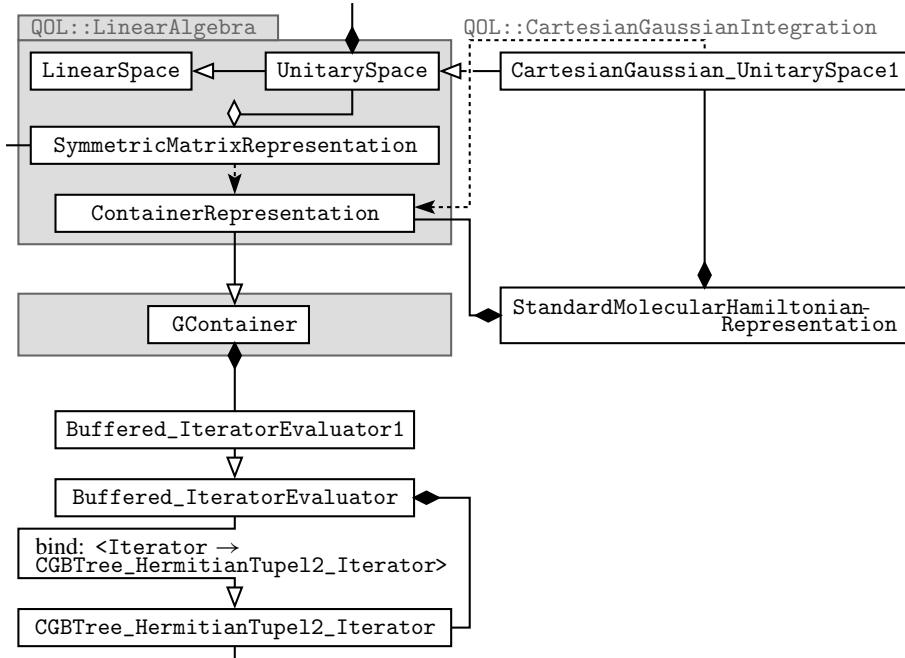


Fig. 4.2: UML class diagrams for parts of the QOL::LinearAlgebra and QOL::CartesianGaussianIntegration namespaces: SymmetricMatrixRepresentation is constructed by ContainerRepresentation that, through the inheritance relationship with the abstract GContainer class, provides the necessary iteration schemes and integral evaluation rules defined by Buffered_IteratorEvaluator1 for the 1-electron operator case.

```

9     return *this;
10 }
  
```

Incrementing the general **Buffered_IteratorEvaluator** increments the special underlying **CGBTTree_HermitianTupel2_Iterator** that, thus, iterates over all CGTO pairs in I . As the flat counter $_i$ runs out of the current batch, $_i$ is reset, and **readAhead()** evaluates the next $_n$ -sized batch, transforms,ⁱ and writes

ⁱ The QOL integral modules can handle (segmented and generalized) contracted CGTO, and CGTO basis sets transformed to spherical-type sets via the respective **CGBTTree_Contracted** and **CGBTTree_Contracted_AngularTransformed** basis set classes. Integral evaluation is, however, im-

the transformed integral batch to the `std::vector<typename Evaluator::value_typeCR>` attribute `_buffer` of `Buffered_IteratorEvaluator`. Then, de-referencing `Buffered_IteratorEvaluator` returns `_buffer`'s element at position `_i`.

The particular implementations of the `readAhead()` methods are, from the nested four-level structure of the CGTO basis set classes `CGBTTree_Contracted` and `CGBTTree_Contracted_AngularTransformed`, somewhat complicated and cannot be discussed in full detail. What is of importance here is that `readAhead()` links QOL iteration and integration by calling the `()` operator of the `Buffered_IteratorEvaluator`'s `Evaluator`-type attribute,^j as briefly discussed below.

1-Electron Integral Evaluation Interface

Generally, all QOL 1-electron operator matrix element integrals are evaluated in terms of integrals over primitive CGTO basis functions, using the interface of the `OneBody_Evaluator` class:

```
1      // typedef typename Operator::_T T;
2 template <class Operator>
3 inline
4 typename OneBody_Evaluator<Operator>::_T
5 OneBody_Evaluator<Operator>::operator () (
6     const CartesianGaussianFunction<T> & bra,
7     const CartesianGaussianFunction<T> & ket ) const {
8
9     return _op(bra, ket);
10 }
```

The `OneBody_Evaluator`'s single template argument `Operator` defines – independently of the `()` operator – the nature of the integrals $O_{pq} = \langle \phi_p | \hat{O} \phi_q \rangle$ to be evaluated, i.e. the 1-electron operator \hat{O} referred to. Moreover, it also defines any

plemented over primitive, non-normalized CGTOs only; integrals over transformed CGTOs are obtained by transforming the primitive integrals. Note, too, that only iteration is done batch-wise, but integral evaluation is not!

^j More precisely, this attribute is of a `const Evaluator &` type, as the actual `Evaluator` object is associated with `StandardMolecularHamiltonianRepresentation`.

particular integral evaluation schemes as implemented in the `Operator` classes' `()` operator called in line 9.

QOL overlap, kinetic energy, nuclear potential energy, and electron–electron repulsion 4-index integrals S_{pq} , T_{pq} , U_{pq} , and g_{pqrs} , are typically evaluated using the Obara–Saika recursion schemes^{122,123} as discussed by Helgaker, Jørgensen, and Olsen.¹⁰¹

Integral Communication and Matrix Representation Construction

Stated naïvely, construction of operator matrix representation objects involves, at first, computation of all (symmetry-unique) integrals and, second, mapping these integrals to the matrix elements in question. Both tasks are effectively addressed within the constructor of, e.g., `SymmetricMatrixRepresentation`, from a `ContainerRepresentation` object by

- converting the `ContainerRepresentation`, i.e. `GContainer` object with `Buffered_IteratorEvaluator1`- and `void`-valued `Iterator` and `Evaluator` template arguments, respectively, to an intermediate `GContainer` with `HermitianRepresentationIterator`- and `MemEvaluator`-valued `Iterator` and `Evaluator` template arguments;
thereby running through the iterator–evaluator hierarchy briefly introduced above;
- mapping and writing of the integrals from the intermediate `GContainer`'s `SymmetricMatrix_MemEvaluator`'s `container` attribute, employing this intermediate `GContainer`'s `HermitianRepresentationIterator`.

The key point is in the difference of the constructor argument `GContainer`'s and the intermediate `GContainer`'s template arguments `Iterator` and `Evaluator`: The former carries the complete iterator–evaluator structure in its `Buffered_IteratorEvaluator1` template argument and attribute, but does not have a `container` attribute allocating any physical memory, i.e. a `void`-type `Evaluator` only. Contrasting, the latter has a simple `HermitianRepresentationIterator` iterator-only structure, but a `MemEvaluator`-type `Evaluator` that does allocate physical memory for the $\frac{1}{2}n(n + 1)$ integrals O_{pq} .

Both GContainer-type objects are linked by their common UnitarySpace-type IndexSpace template argument. Stated explicitly,

```
1 template <class value_type, class IndexSpace,
2           class Iterator1, class Evaluator>
3 template <class Iterator2>
4 inline
5 GContainer<value_type, IndexSpace, Iterator1, Evaluator>:::
6 GContainer(const GContainer<
7   value_type, IndexSpace, Iterator2, void> & g1) :
8   // ...
9 {
10   for( Iterator2 I(g1.indexSpace(), g1.evaluator());
11        I.valid(); ++I )
12     (*this)[I.operator Iterator1()] = g1[I];
13 }
```

As Iterator2 is Buffered_IteratorEvaluator1-valued, the loop in lines 10–12 runs through the QOL iterator-evaluator hierarchy, thereby evaluating the integrals through the coupling of `readAhead()` to the `++` operator as discussed above. If the Evaluator template argument is `void`-valued, as it is for the constructor argument `g1`, the `[]` operator de-references `g1`'s `Buffered_IteratorEvaluator` attribute and returns the integral corresponding to `I`'s current position. In line 12, this integral is assigned to the `MemEvaluator`'s container attribute, accessed via the `[]` operator, at the position defined by the type conversion operator from `Buffered_IteratorEvaluator1`'s base class `CGBTREE_HermitianTupel2_Iterator` to `HermitianRepresentationIterator`.

Note that it is this conversion of the complicated, 8-loop `Buffered_IteratorEvaluator1` iteration scheme to the simple iteration scheme of `HermitianRepresentationIterator` that maps $\{O_{pq}\}_{pq}$ to the sequence $(O_{pq})_{p \geq q}$ readily cast to, e.g., the lower triangular part of the matrix container class: Within the constructor of `SymmetricMatrixRepresentation` from `ContainerRepresentation`, i.e. `GContainer`,

```
1 // template <class CR>
```

^k Note that, in the type definition in lines 3–8, the fourth template argument is not stated explicitly, and is thus set to its default value `MemEvaluator<typename CR::UnitarySpace, HermitianRepresentationIterator<typename CR::UnitarySpace>>`.

```
2      // ...
3  typedef GContainer<
4      typename CR::UnitarySpace::_Field,
5      typename CR::UnitarySpace,
6      HermitianRepresentationIterator<typename
7          CR::UnitarySpace>
8      > G2;
9
10 _Field * P = &(*this)(0,0);
11 G2 g2(g1);
12
13 int j = 0;
14 for( typename G2::const_iterator I(g2); I.valid(); ++I )
15     P[j++] = g2[I];
```

In line 10, P is set to the address of the first element of SymmetricMatrix_Mem-Evaluator's container; the loop in lines 14–15 runs within the HermitianRepresentationIterator's (“horizontal”-lower triangular) iteration scheme.

4.3.3 The QOL Self-Consistent Field Algorithm

The 2006 QOL implementation's self-consistent field algorithm is a closed-shell, integral-conventional, null-guess, straightforward implementation as discussed by, e.g., Szabo and Ostlund¹⁰⁰ or Almlöf.¹⁰⁵

Stated briefly, parsing molecular structure and CGTO basis set information by constructing the respective Molecule and StandardMolecularHamiltonian-Representation class objects is followed by

- construction of **S**, **T**, and **U** matrices;
- evaluation of all symmetry-unique electron–electron repulsion 4-index integrals $g_{prqs} \in \mathcal{G}_n$, and storage;
- setting the initial guess density matrix **D**⁰ to null;
then, for $\delta_{\mathbf{D}} > 0$ a density threshold, in the μ -th SCF iteration

1: assemble $\mathbf{G}^\mu = \mathbf{G}(\mathbf{D}^\mu)$ from \mathbf{D}^μ via eq. 3-45;¹

¹ The discussion of the construction of matrix representation class objects from ContainerRepresentation given in sec. 4.3.2 does not apply directly to the $\mathbf{G} = \mathbf{J} + \mathbf{K}$ matrix: It is clearly more

- 2: assemble the μ -th Fock matrix $\mathbf{F}^\mu = \mathbf{F}(\mathbf{D}^\mu) = \mathbf{h} + \mathbf{G}^\mu$;
 - 3: orthogonalize \mathbf{F}^μ , diagonalize, and back-transform to obtain the μ -th MO eigenvector matrix \mathbf{C}^μ ;
 - 4: assemble a new density matrix $D_{pq}^{\mu+1} = 2 \sum_{i=1}^N C_{pi}^\mu C_{iq}^{\mu*}$ from the μ -th vector representations of the N doubly “occupied”, i.e. the N lowest-energy MOs;
 - 5: terminate if $\|\mathbf{D}^{\mu+1} - \mathbf{D}^\mu\|_2 \leq \delta_{\mathbf{D}}$; else go to 1.
- computation of the Hartree–Fock SCF energy $E_{\text{SCF}}(\mathbf{D}^{\mu+1})$ via $E_{\text{SCF}} = \frac{1}{2} \sum_{pq} D_{pq}^{\mu+1} (2h_{pq} + J_{pq} - K_{pq})$

efficient to iterate over the $\mathcal{O}(n^4)$ unique $g_{prqs} \in \mathcal{G}_n$, because every g_{prqs} contributes to a large part of \mathbf{G} matrix elements $G_{pq} = \sum_{rs} D_{sr} (2g_{prqs} - g_{prsq})$, i.e. to choose an “integral-driven” assembly of \mathbf{G} .

The 2006 QOL implementation provides a global function `assembleRHFMatrix_2BodyPart2` that returns a `SymmetricMatrixRepresentation` from `CartesianGaussian_UnitarySpace-2`, `SymmetricMatrixRepresentation`, and `std::vector<double>` arguments corresponding to $\mathbb{B} \otimes \mathbb{B}$, \mathbf{D} , and \mathcal{G}_n , respectively. Technically, `assembleRHFMatrix_2BodyPart2` employs a `CGBTTree_HermitianTupel4_Iterator` class to run over all input $g_{prqs} \in \mathcal{G}_n$; every given unique g_{prqs} is then added to all \mathbf{J} and \mathbf{K} matrix elements it contributes to (cf. sec. 7.1).

5

Matrix Algebra

The 2-component Roothaan–Hall- SCF matrix equations discussed in sec. 3.2, i.e.

$$\mathbf{FC} = \mathbf{SC}\epsilon,$$

differ from their spin-free analog of eq. 3-44 in two main points: First, because of the non-vanishing spin-orbit pseudopotential integrals W_{pq} the Fock matrix \mathbf{F} and, thus, its eigenvector matrix \mathbf{C} are generally complex-valued. Second, \mathbf{F} does not decouple as $\mathbf{F}_{\alpha\alpha} \oplus \mathbf{F}_{\beta\beta}$ from spin symmetry breaking, and one has to consider the full $\mathbb{C}^{2n \times 2n}$ problem.^a

From their abstract and general structure of inheritance hierarchies of template classes the QOL matrix and matrix representation algebra modules are, in principle, well suited for the generalization to the spin symmetry-broken 2-component framework. However, due to the simpler problem setting of the spin-free eq. 3-44, the 2006 QOL implementation certainly did not exploit the full flexibility it could have from its general structure. Therefore, outside new classes that had to be added, a number of changes had to be made to the existing matrix and matrix representation algebra modules to fully integrate the new components.

The presentation in this chapter follows both the general bottom-up strategy

^a This is true only if one restricts the decoupling transformation to be complex-valued. For time reversal-invariant \mathbf{F} , one can find quaternion-valued decoupling transformations and solve time reversal-related smaller $\mathbb{H}^{n \times n}$ problems.¹¹¹ However, this strategy has not been employed in this work because of the difficulties arising from the need for efficient $\mathbb{H}^{n \times n}$ equation solving.

and the concepts of the 2006 QOL implementation discussed in sec. 4.2.2: Matrix algebra, matrix representation algebra, and equation solving has been kept separate and is discussed in this order. Time reversal-invariance was introduced on the lowest possible, i.e. the matrix container class level as abstract block symmetry, providing ready-made interfaces for quaternion algebra modules. Complexification was introduced by abstract inner products connecting complex-valued matrix with top-level matrix representation classes through formally complex codomains.

5.1 Matrix Container and Algebra Classes

The two additional class hierarchies for complex-valued matrix algebra, providing base classes `HermitianMatrix` and `HermitianCayleyMatrix` for deriving top-level hermitian and hermitian time reversal-invariant matrix representation classes, respectively, follow the concepts as already discussed for the `SymmetricMatrix` class in sec. 4.3.1. The following discussion will, therefore, be limited to the most important ideas and implementation details only.

5.1.1 Hermitian and Time Reversal-Like Block Symmetry

Within the `HermitianMatrix` class hierarchy analog to fig. 4.1, the container-evaluator class `HermitianMatrix_MemEvaluator` differs from `SymmetricMatrix_MemEvaluator` in the `value_type` template argument, which is assumed to be of a `std::complex` type. Moreover, symmetry-redundant matrix elements $M_{pq} \notin \mathbb{R}$ are generated from symmetry-unique ones differently: Clearly, storing only symmetry-unique matrix elements implies read access via

```
1 value_type operator () (int p, int q) const {
2     if( matrixPackageOrder == LowerDiagonal ) {
3         if( p < q ) {
4             std::swap(p,q); return conj( _v[p*(p+1)/2 + q] );
5         }
6         else return _v[p*(p+1)/2 + q];
7     } // ...
```

i.e. returning $M_{qp}^* \neq M_{pq}$ for $i < j$. This is, however, payed for by the two-index () operator returning copies, and by the need for a separate write access method

```
void setValue(int p, int q, const value_type & v).
```

Implementation of the HermitianMatrix class involves only a small number of methods to overload inherited Matrix functions, e.g. norm2() returning $\|\mathbf{M}\|_2 = \sqrt{\sum_{pq} |M_{pq}|^2}$ in place of $\sqrt{\sum_{pq} M_{pq}^2}$.

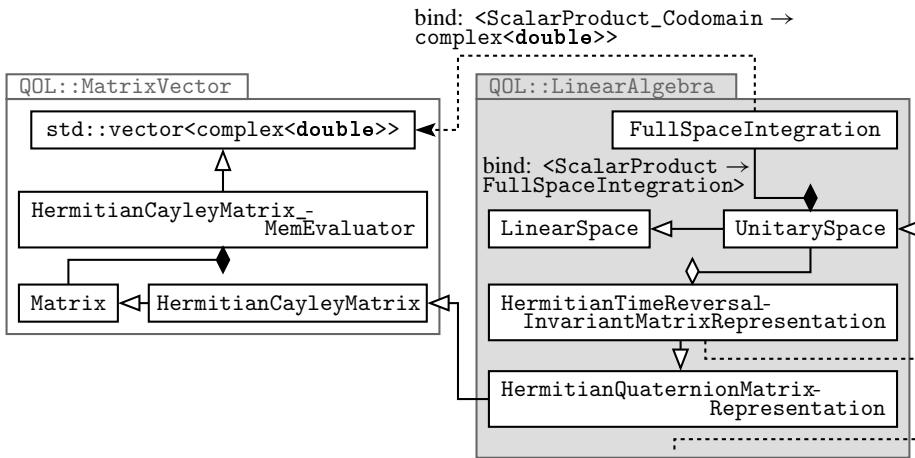


Fig. 5.1: UML class diagrams for the new HermitianCayleyMatrix and HermitianTimeReversalInvariantMatrixRepresentation classes: All template parameters and most implied template argument bindings have been omitted for clarity.

As time reversal symmetry cannot be meaningfully referred to on an abstract, i.e. matrix algebra-only level without referring to the concept of matrix representations of time reversal-invariant operators, the special structure of eq.s 3-34 has been introduced through the concept of “Cayley matrices”: A $\mathbb{C}^{2n \times 2n}$ matrix \mathbf{M} of the special “time reversal-like” symmetry type

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{\alpha\alpha} & \mathbf{M}_{\alpha\beta} \\ \mathbf{M}_{\beta\alpha} & \mathbf{M}_{\beta\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{\alpha\alpha} & \mathbf{M}_{\alpha\beta} \\ -\mathbf{M}_{\alpha\beta}^* & \mathbf{M}_{\alpha\alpha}^* \end{pmatrix} \quad (5-1)$$

for $\mathbf{M}_{\alpha\alpha}, \mathbf{M}_{\alpha\beta} \in \mathbb{C}^{n \times n}$ is referred to as “Cayley matrix” or “being of Cayley symmetry”.^b For hermitian Cayley matrices \mathbf{M} , hermiticity introduces additional

^b “Cayley matrix” is a reference to the British mathematician Arthur Cayley, 1821–1895, and his name’s connection to the Cayley–Dickson construction^{124,125} of a sequence of algebras $\mathbb{A}_{i+1} =$

structure by restricting $\mathbf{M}_{\alpha\alpha}$ and $\mathbf{M}_{\alpha\beta}$ to be hermitian and antisymmetric, respectively:

$$M_{pq\gamma\gamma} = M_{qp\gamma\gamma}^*, \quad M_{pq\gamma'\gamma''} = -M_{pq\gamma''\gamma'}^* = -M_{qp\gamma'\gamma''} \quad (5-2)$$

for $\gamma, \gamma', \gamma'' \in \{\alpha, \beta\}$, $\gamma' \neq \gamma''$.

The HermitianCayleyMatrix class hierarchy is similar to the HermitianMatrix case. From the richer internal structure given by eq.s 5-2, however, storage of only the $\frac{1}{2}n(n+1) + \frac{1}{2}n(n-1) = n^2$ symmetry-unique matrix elements, and access-related index operations, are more involved. The container-evaluator class HermitianCayleyMatrix_MemEvaluator stores either the $\mathbf{M}_{\alpha\alpha}$ upper and $\mathbf{M}_{\alpha\beta}$ lower triangular part, or the $\mathbf{M}_{\alpha\alpha}$ lower and $\mathbf{M}_{\beta\alpha}$ upper triangular part, respectively, excluding the $\gamma \neq \gamma'$ block's vanishing diagonal, as controlled by its template parameter HermitianCayleyMatrixPackageOrder and illustrated in fig. 5.2 Again, read and write access requires to return copies from the two-

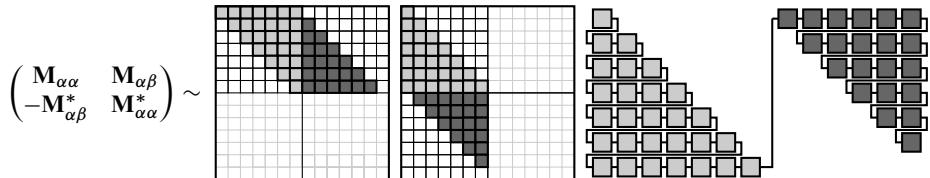


Fig. 5.2: Layout of the HermitianCayleyMatrix_MemEvaluator's STL vector-type container class: Only n^2 of $4n^2$ matrix elements are stored in a horizontal-“dictionary” (left) or horizontal-“sequential” (right) fashion; the latter is more convenient for implementation purposes as the $\mathbf{M}_{\alpha\alpha}$ block is given by the first $\frac{1}{2}n(n+1)$ consecutive container elements.

index () operator, symmetry-redundant matrix elements being generated from symmetry-unique ones by eq.s 5-2, and a separate **void setValue(int p, int q, const value_type & v)** method, respectively.

The implementation of HermitianCayleyMatrix has been done closely analogous to HermitianMatrix.

$\mathbb{A}_i \oplus \mathbb{A}_i$. The \mathbb{A}_{i+1} algebra's elements have an \mathbb{A}_i -valued 2×2 matrix-like array representation of the special structure given by eq. 5-1. In fact, \mathbf{M} of eq. 5-1 is a representation of a quaternion-valued $n \times n$ matrix¹²⁶ as discussed in sec. 5.1.2.

5.1.2 Quaternion Matrix Matrix Representations

The $\mathbb{C}^{2n \times 2n}$ matrices of eq. 5-1 can be interpreted as complex-valued matrix representations of quaternion-valued $n \times n$ matrices that, through their block symmetry, mimick non-commutative quaternion multiplication.¹²⁶ Whereas there is, in principle, no need to refer to this abstract algebraic connection from the point of theory, its use¹¹¹ has been shown to simplify computations in the Kramers restricted HF and DHF SCF¹²⁷ and MCSCF¹²⁸ frameworks. This connection will be briefly reviewed to address the HermitianQuaternionMatrixRepresentation auxiliary class implementation.

Any quaternion^c square matrix \mathbf{Y} can be identified with a pair of complex-valued matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n \times n} \subset \mathbb{H}^{n \times n}$ as $\mathbf{Y} = \mathbf{A} + \mathbf{B}\mathbf{j}$. This is a linear bijective map¹¹¹

$$\begin{aligned} \mathcal{N}: \mathbb{C}^{n \times n} \times \mathbb{C}^{n \times n} &\rightarrow \mathbb{H}^{n \times n}, \\ (\mathbf{A}, \mathbf{B}) &\mapsto \mathbf{Y} = \mathbf{A} + \mathbf{B}\mathbf{j} = \operatorname{Re} \mathbf{A} + i\operatorname{Im} \mathbf{A} + j\operatorname{Re} \mathbf{B} + i \cdot j\operatorname{Im} \mathbf{B}. \\ &= \mathbf{A} + \mathbf{B}\mathbf{j} = \operatorname{Re} \mathbf{A} + i\operatorname{Im} \mathbf{A} + j\operatorname{Re} \mathbf{B} + k\operatorname{Im} \mathbf{B} \end{aligned} \quad (5-3)$$

Thus, for quaternion square matrices $\mathbf{Y} = \mathbf{A} + \mathbf{B}\mathbf{j}$ and $\mathbf{Z} = \mathbf{C} + \mathbf{D}\mathbf{j}$, noting $\mathbf{A}\mathbf{j} = \mathbf{j}\mathbf{A}^*$,

$$\mathbf{Y}\mathbf{Z} = (\mathbf{A} + \mathbf{B}\mathbf{j})(\mathbf{C} + \mathbf{D}\mathbf{j}) = (\mathbf{AC} - \mathbf{BD}^*) + (\mathbf{AD} + \mathbf{BC}^*)\mathbf{j}, \quad (5-4)$$

$$\mathbf{Y}^* = \mathbf{A}^* - \mathbf{j}\mathbf{B}^* = \mathbf{A}^* - \mathbf{B}\mathbf{j}. \quad (5-5)$$

It is easily established from eq.s 5-4 and 5-5 that the bijective map

$$\mathcal{M}: \mathbb{H}^{n \times n} \rightarrow \mathbb{A}_n \subset \mathbb{C}^{2n \times 2n}, \quad \mathbf{A} + \mathbf{B}\mathbf{j} \mapsto \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{B}^* & \mathbf{A}^* \end{pmatrix} \ni \mathbb{A}_n \quad (5-6)$$

^c Stated very briefly, the quaternions \mathbb{H} are numbers $Q = a + ib + jc + kd$ with $a, b, c, d \in \mathbb{R}$ and unit products

$$i^2 = j^2 = k^2 = ijk = -1 \Rightarrow ij = k \text{ and cyclic,}$$

respecting the order of factors, and with quaternion conjugation and norm-square defined by $Q^* := a - ib - jc - kd$ and $|Q|^2 := QQ^* = Q^*Q = a^2 + b^2 + c^2 + d^2$, respectively. \mathbb{H} is a non-commutative normed division algebra, and $\mathbb{H} \subset \mathbb{C} \subset \mathbb{R}$. The importance of quaternions comes from the possibility to provide a unified calculus for classical, special- and general-relativistic, and quantum theories through Clifford algebras (which can be constructed as tensor products of quaternion algebras).¹²⁶

is an isomorphism of the quaternion matrix algebra $\mathbb{H}^{n \times n}$ and the sub-algebra \mathbb{A}_n of complex-valued non-singular $2n \times 2n$ matrices of the special type defined by eq. 5-6, with quaternion multiplication and conjugation represented by matrix multiplication and taking the hermitian adjoint, respectively.¹¹¹

With \mathcal{M}^{-1} defined by inverting eq. 5-6, matrix representations of hermitian time reversal-invariant operators can be interpreted as hermitian quaternion-valued matrices, and algebraic manipulation can be carried out in $\mathbb{H}^{n \times n}$ – which is possibly favorable for steep scaling algorithms –, following mapping to $\mathbb{C}^{2n \times 2n}$ via \mathcal{M} .¹¹¹

Every complex-valued matrix representation $\mathcal{M}(\mathbf{Y})$ of a quaternion matrix \mathbf{Y} is, from eq. 5-6, a Cayley matrix by eq. 5-1. However, the converse is not true; from eq. 5-1 there is no need to refer to the algebra isomorphism \mathcal{M} , and eq. 5-1 does not restrict a non-zero Cayley Matrix \mathbf{M} to be non-singular. This relationship is easily implemented as inheritance, i.e. by deriving the auxiliary class `HermitianQuaternionMatrixRepresentation` from its base class `HermitianCayleyMatrix` (and contracting the somewhat clumsy expression “quaternion matrix matrix representation” to “quaternion matrix representation”).

Within this structure one can easily connect complex- and quaternion matrix algebra in a transparent and meaningful way. For example, the constructor

```
1 template <MatrixPackageOrder mP0, class Evaluator>
2 HermitianQuaternionMatrixRepresentation(
3     const HermitianMatrix<
4         QOL::ElementaryMath::Quaternion<typename
5             value_type::value_type>, mP0, Evaluator> & );
```

constructs a `HermitianQuaternionMatrixRepresentation`-type object from a `QOL::ElementaryMath::Quaternion`-valued `HermitianMatrix` object via \mathcal{M} , and plays an important role in the QOL PP integral communication discussed in ch. 6.

5.2 Matrix Representation Classes

A straightforward implementation of matrix representation class hierarchies analog to `SymmetricMatrixRepresentation` discussed in sec. 4.3.1 is spoiled by

the 2006 QOL implementation’s connection of the matrix representation elements’ codomain – which is restricted to \mathbb{R} – to the CGTO basis functions these are evaluated from. This link is, as discussed in sec. 4.3.1, provided by the `UnitarySpace` and `ScalarProduct` classes, which thus had to be modified to enable implementation of `HermitianMatrixRepresentation` and `HermitianTimeReversalInvariantMatrixRepresentation` class hierarchies coherent with the 2006 QOL implementation.

5.2.1 QOL Unitary Spaces and Scalar Products

The 2006 QOL implementation of the matrix representation modules discussed in sec. 4.3.1 cannot handle spin–orbit (pseudopotential) operator matrix elements. This is because of the complex-valued representation of the spin–orbit PP operators \hat{B} of eq. 2-48 in the CGTO basis: For a $\mathbb{R}^3 \rightarrow \mathbb{R}$ CGTO $|\phi\rangle \in \mathbb{B}$ and $\gamma, \gamma' \in \{\alpha, \beta\}$, generally $\langle \mathbf{r}\gamma | \hat{B}\phi\gamma' \rangle \notin \mathbb{R}$, such that $\langle \gamma | \hat{B}\phi\gamma' \rangle \notin \mathbb{B}$. In other words, whereas the integral exists, $\langle \phi_p \gamma | \hat{B}\phi_q \gamma' \rangle$ is not a $\mathbb{B} \times \mathbb{B}$ scalar product, thus clearly not property of only \mathbb{B} .

Technically, this is intimately connected to the definition of the matrix representation classes’ `value_type` template parameter as `UnitarySpace::ScalarProduct::_Field`, i.e.

```
1 template <class Vector>
2 class ScalarProduct {
3   public: typedef typename Vector::_Field _Field;
```

which defines matrix representation elements to lie in a $\mathbb{B} \times \mathbb{B}$ scalar product codomain, and identifies this scalar product codomain with the CGTOs’ domain of definition `Vector::_Field`.

To overcome this limitation, but maintaining the 2006 QOL implementation’s global structure for compatibility reasons, the CGTO-spanned space \mathbb{B} was “extended” to include $\langle \gamma | \hat{W}\phi\gamma' \rangle$. This “extension” of \mathbb{B} is its complexification.

Formal Concept: Finite-Dimensional Function Space Complexification

Let $\mathbb{B} = \text{span}(|\phi_p\rangle)_{p=1}^n$ be the \mathbb{R} -function space spanned by $n \mathbb{R}^3 \rightarrow \mathbb{R}$ CGTOs $|\phi_p\rangle$. Its complexification \mathbb{B}' is the tensor product of \mathbb{C} – seen as 2-dimensional

\mathbb{R} -space – with \mathbb{B} ,¹²⁹ i.e. the function space

$$\mathbb{B}' := \mathbb{C} \otimes \mathbb{B} = \text{span} \left((1 \otimes |\phi_p\rangle)_{p=1}^n \cup (i \otimes |\phi_p\rangle)_{p=1}^n \right). \quad (5-7)$$

Then, any $|\phi\rangle \in \mathbb{B}'$ has a unique decomposition in terms of the basis functions of \mathbb{B}' ,

$$|\phi\rangle = \sum_p a_p (1 \otimes |\phi_p\rangle) + \sum_p b_p (i \otimes |\phi_p\rangle) = \sum_p y_p \otimes |\phi_p\rangle, \quad (5-8)$$

with $a_p, b_p \in \mathbb{R}$ and $y_p = a_p + ib_p$, from the linearity of the tensor product. One can, then, define multiplication of complex numbers z with functions in $|\phi\rangle \in \mathbb{B}'$ by

$$(z, |\phi\rangle) \mapsto \sum_p (z \cdot y_p) \otimes |\phi_p\rangle =: \sum_p (z \cdot y_p) |\phi'_p\rangle = z |\phi\rangle \quad (5-9)$$

such that $\mathbb{B}' = \mathbb{C} \otimes \mathbb{B}$ becomes an n -dimensional \mathbb{C} -function space with basis functions $|\phi'_p\rangle := 1 \otimes |\phi_p\rangle$, i.e. the space of complex-valued linear combinations of $\mathbb{R}^3 \rightarrow \mathbb{R}$ CGTOs. The structure of \mathbb{B} is retained as \mathbb{B} is a proper “real-valued” subspace of \mathbb{B}' .

From

$$\langle \phi'_q | \phi \rangle = \sum_p y_p \langle \phi'_q | \phi'_p \rangle = y_q \in \mathbb{C} \quad (5-10)$$

since $\langle \phi'_q | \phi'_p \rangle = \langle 1 \otimes \phi_q | 1 \otimes \phi_p \rangle = \delta_{qp}$, \mathbb{B}' can be considered to have a $\mathbb{B}' \times \mathbb{B}'$ scalar product implied by its “real-valued” subspace’s $\mathbb{B} \times \mathbb{B}$ scalar product, e.g. integration over \mathbb{R}^3 , but with a complex-valued codomain.

Separation of Vector::Field and The Scalar Product’s Codomain

As given through eq. 5-10, complexification of the CGTO-spanned function space \mathbb{B} within the 2006 QOL implementation’s framework has been achieved by separating the CGTO’s domain of definition from the CGTO-spanned space’s scalar product codomain, and assigning a `std::complex` type to the latter.

To maintain compatibility with the existing matrix representation and 1-component HF SCF modules, the `CartesianGaussian_UnitarySpace1` template class was given a second template parameter `ScalarProduct_Codomain`:

Matrix Representation Classes

```
1 /* ----- STATUS QUO END 2006 QOL IMPLEMENTATION
2
3 template <class CGTOBasis>
4 CartesianGaussian_UnitarySpace1 :
5   public UnitarySpace<LinearSpace<CGTOBasis>,
6     FullSpaceIntegration<typename CGTOBasis::_T>>
7 {
8
9
10 template <class CGTOBasis,
11   class ScalarProduct_Codomain = typename CGTOBasis::_T>
12 CartesianGaussian_UnitarySpace1 :
13   public UnitarySpace<LinearSpace<CGTOBasis>,
14     FullSpaceIntegration<ScalarProduct_Codomain>>
15 }
```

Whereas this decouples the CGTO basis $(|\phi_p\rangle)_p$ and the CGTO-spanned space's scalar product codomain completely, the 2006 QOL implementation remains untouched through the default value of the second template parameter to the CGTO domain of definition, effectively recovering the special case of a $\mathbb{B} \times \mathbb{B} \rightarrow \mathbb{R} \subset \mathbb{C}$ scalar product.^d

Unitary “Representation” and “Index” Space Decoupling

As discussed in sec.s 4.3.1 and 4.3.2, construction of matrix representation class objects from the `StandardMolecularHamiltonianRepresentation` classes' `ContainerRepresentation` attributes is linked to QOL iteration, integration, and integral communication through the CGTO-spanned function space: The corresponding `CartesianGaussian_UnitarySpace1` class is both a template argument of the matrix representation class and an attribute of `StandardMolecularHamiltonianRepresentation`, being communicated by `GContainer` as its template argument `IndexSpace`. Therefore, construction of `HermitianTimeReversalInvariantMatrixRepresentation` object should involve communication

^d In an analog way, the `ScalarProduct` class was changed from a 1-parameter template to become a 2-parameter template class, the second template parameter being a `class ScalarProduct_Codomain` set to the default `typename Vector::Field`.

of a `CartesianGaussian_UnitarySpace1` object with a `ScalarProduct_Co-domain` type definition corresponding to the complex-valued regime.

However, the spin-independent operators \hat{T} , \hat{U} , and the spin-free pseudo-potential operator \hat{A} , have real-valued matrix representations \mathbf{O} that, moreover, decouple as $\mathbf{O} = \mathbf{O}_{\alpha\alpha} \oplus \mathbf{O}_{\beta\beta}$, with the identity $\mathbf{O}_{\beta\beta} = \mathbf{O}_{\alpha\alpha}^* = \mathbf{O}_{\alpha\alpha} \in \mathbb{R}^{n \times n}$ from time reversal symmetry.^e These block matrices are the same as in a spin-free 1-component HF SCF calculation scheme. Therefore, it is not necessary to adapt the 2006 QOL implementation’s iterator–evaluator structure to the complex-valued regime. In fact, it is possible to make use of the established structure for the construction of \mathbf{T} , \mathbf{U} , and \mathbf{A} by choosing the “real-valued” subspace $\mathbb{B} \subset \mathbb{B}'$ for these block matrices; only the spin–orbit part \hat{B} of \hat{W} needs the full \mathbb{B}' for its representation. Effectively this means to decouple the unitary “representation” space referred to for operator matrix representation, i.e. generally \mathbb{B}' , from the unitary “index” space the iterator–evaluator structure uses for iteration, integration, and integral communication.^f

The top-level `StandardMolecularHamiltonianRepresentation` class was, thus, replaced by the essentially analog 1-component and 2-component HF SCF calculation scheme classes

```
1 /* ----- 1-COMPONENT HF SCF CALCULATION SCHEME ----- */
2 template <class CGTOBasis>
3 class
4 StandardMolecularHamiltonianOrbitalRepresentation_Container
5 {
6 public:
7     typedef CartesianGaussian_UnitarySpace1<CGTOBasis>
8         USpace;
9     typedef CartesianGaussian_UnitarySpace1<CGTOBasis>
10        U1Space1;
```

^e This is also true for \hat{J} . The exchange operator \hat{K} is, however, not spin-independent as it involves permutations of both spatial and spin electron coordinates.

^f As discussed in ch. 6, the ARGOS PP integral routines compute both $W_{pq\alpha\alpha}$ and $W_{\alpha\beta}$ integrals in a single call from the respective CGTOs $|\phi_p\rangle$ and $|\phi_q\rangle$, permitting a single iteration over $I \in B \times B$ for the construction of both PP matrix representation blocks $\mathbf{W}_{\alpha\alpha}$ and $\mathbf{W}_{\alpha\beta}$.

Note that, formally, the “index” space does not need the inner product information and should be of the base class type `LinearSpace` instead of the derived `UnitarySpace`.

and

```
1 /* ----- 2-COMPONENT HF SCF CALCULATION SCHEME ----- */
2 template <class CGTOBasis>
3 class
4 StandardMolecularHamiltonian2SpinorRepresentation_Container
5 {
6 public:
7     typedef CartesianGaussian_UnitarySpace1<CGTOBasis,
8         std::complex<typename CGTOBasis::_T>> USpace;
9     typedef CartesianGaussian_UnitarySpace1<CGTOBasis>
10        U1Space1;
```

respectively. As opposed to the 2006 QOL implementation, both have two distinct `CartesianGaussian_UnitarySpace1` type definitions and attributes. The first, defined in lines 7–8, set the global CGTO-spanned “representation” space \mathbb{B}' communicated externally to matrix representation class objects to be constructed; the second, defined in lines 9–10, define a “index” space $\mathbb{B} \subseteq \mathbb{B}'$ communicated internally to the iterator–evaluator structures.

This construction leaves large parts of the 2006 QOL implementation, particularly the complicated iterator–evaluator structure, essentially unchanged, and provides very similar class interfaces for both 1-component and 2-component HF SCF calculation schemes. However, it necessarily requires to communicate a unitary “representation” space object to the matrix representation classes’ constructors from `GContainer`, because this is generally different from the `GContainer`’s unitary “index” space.^g

5.2.2 Symmetric, Hermitian, and Hermitian Time Reversal-Invariant Matrix Representations

Both the `HermitianMatrixRepresentation` and `HermitianTimeReversalInvariantMatrixRepresentation` class hierarchies have been designed ana-

^g This is the single substantial change to the 2006 QOL implementation’s class structure interface. In principle, one did not need to hand over a unitary “representation” space object to the `SymmetricMatrixRepresentation` constructor from `GContainer`, because it coincides with the latter’s unitary “index” space. However, the `SymmetricMatrixRepresentation` constructors have been modified accordingly to provide a larger measure of similarity among the 1-component and 2-component scheme interfaces.

log to the `SymmetricMatrixRepresentation` class hierarchy established with the 2006 QOL implementation, i.e. have been built on the `HermitianMatrix` and `HermitianCayleyMatrix` class hierarchies, respectively. Fig. 5.1 illustrates the class structure for the `HermitianTimeReversalInvariantMatrixRepresentation` case

The `HermitianTimeReversalInvariantMatrixRepresentation` hierarchy differs from this general structure in its auxiliary (base) class `HermitianQuaternionMatrixRepresentation`. This construction allows both quaternion algebra operations as, e.g., to address only the spin-orbit part of $\mathbf{O}_{\alpha\alpha}$ and $\mathbf{O}_{\alpha\beta}$ by accessing the imaginary parts of the corresponding quaternion matrix $\mathcal{M}^{-1}(\mathbf{O})$, by inheritance of `HermitianQuaternionMatrixRepresentation` methods and operators, and construction from QOL::ElementaryMath::Quaternion-valued `HermitianMatrix` objects as discussed in sec. 5.1.2.

Both the `HermitianMatrixRepresentation` and `HermitianTimeReversalInvariantMatrixRepresentation` classes have a constructor from `UnitarySpace` and `GContainer` analog to that of `SymmetricMatrixRepresentation`. The latter has, however, been changed from the need to decouple unitary “representation” and “index” spaces, i.e.

```
1 /* ----- STATUS QUO END 2006 QOL IMPLEMENTATION
2
3 template <class Iterator, class Evaluator>
4 SymmetricMatrixRepresentation(
5     const GContainer<
6         typename UnitarySpace::ScalarProduct::_Field,
7         UnitarySpace, Iterator, Evaluator> & );
8
9
10 template <class GC>
11 SymmetricMatrixRepresentation( const UnitarySpace &,
12     const GC & );
```

as discussed in the preceding section.

It is through this construction that it is possible to make use of the unchanged 2006 QOL implementation’s iterator–evaluator structure to construct the **S**, **T**, and **U** matrices – provided a horizontal–“sequential” layout of the underlying `HermitianCayleyMatrix_MemEvaluator`’s container, as illustrated in fig.s 5.2

and 5.3, is chosen. Then, the QOL iterator–evaluator structure computes $\text{ord } I = \frac{1}{2}n(n + 1)$ matrix representation elements $\{O_{pq}\}_{pq}$; by HermitianRepresentationIterator, $(O_{pq})_{p \geq q}$ is then mapped directly to the first $\frac{1}{2}n(n + 1)$ HermitianCayleyMatrix_MemEvaluator’s container elements, which are associated with the non-vanishing $\mathbf{O}_{\alpha\alpha}$ block, thus leaving the elements associated with $\mathbf{O}_{\beta\alpha} = \mathbf{0}_n$ unchanged.^h

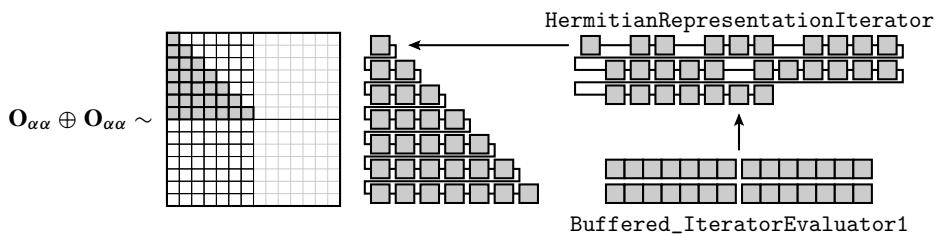


Fig. 5.3: Construction of spin-free operator matrix representations within a 2-component scheme: The $\frac{1}{2}n(n + 1)$ integrals $(O_{pq})_{p \geq q}$ map to the lower triangular part of the symmetric $n \times n$ blocks $\mathbf{O}_{\alpha\alpha}$ through the HermitianRepresentationIterator’s lower triangular “horizontal” iteration scheme.

5.3 Output Formatting and Visualization

Output, i.e. `<<` operators for `HermitianMatrix` and `HermitianCayleyMatrix` – and, by type downcast, for the matrix representation classes of sec. 5.2.2 – have been implemented analog to that of `SymmetricMatrix`. Thus, both use the `QOL::IO::AlignedArray` class that converts matrix elements to (character) strings and, depending on its `AlignMode` attribute, hands a set of left-, center-, right-, or decimal point-justified matrix element strings to a `ostream` reference via the `AlignedArray`’s `<<` operator.

^h Within a 1-component HF SCF calculation scheme only, this was also true for the spin-free PP operator matrix representation \mathbf{A} . Including this as a special case of the more general 2-component HF SCF calculation scheme, however, requires an altogether different implementation. Stated briefly – and discussed in more detail in sec. 6.3.3 –, construction of the \mathbf{W} matrix involves a constructor specialization that, first, constructs a quaternion-valued hermitian matrix from $W_{pq\alpha\alpha}$ and $W_{pq\alpha\beta}$ via the $\mathbb{C} \times \mathbb{C} \rightarrow \mathbb{H}$ map \mathcal{N} , eq. 5-3, and, second, maps this to the complex-valued $2n \times 2n$ matrix \mathbf{W} via \mathcal{M}^{-1} .

For the `std::complex-valued HermitianMatrix` and `HermitianCayleyMatrix` classes a special `AlignMode` enumeration type `DecimalPoint_Suited4Complex` was implemented. `DecimalPoint_Suited4Complex` effects decimal point-justification for real and imaginary parts, and a canonical complex number notation, i.e.

```
[ -1.98 + 7.83 i      6.29 + 2.78 i      7.17 -95.2 i
 -80.4 + 0.163i     -2.18 +13. i      -0.524+61.3 i
 -7.7 + 2.93 i      9.19 - 0.283i     -6.63 +52.6 i ]
```

This format is GNU Octave-compatible and can thus conveniently be processed also outside the QOL.

Within the `QOL::IO` namespace, a global 1-parameter template function

```
1 template <class T>
2 void Matrix2Fig( const Matrix<T> &, ostream & );
```

has been implemented to enable simple visualization, e.g. for easy inspection of matrix structure or symmetry, as showcased in fig. 5.4. Generally, graphical data

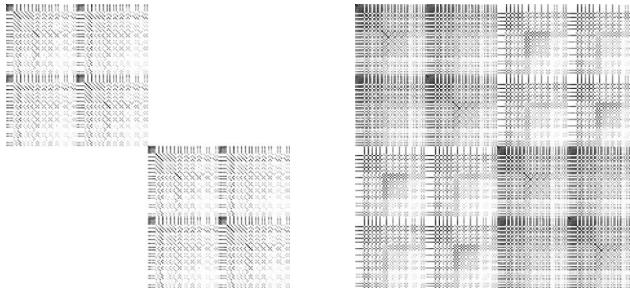


Fig. 5.4: Matrix visualization: 2-component HF SCF density matrices for Rn_2 at $d_{\text{Rn-Rn}} = 7 \text{ a.u.}$, small-core MC-DHF PP with aug-cc-pVTZ basis, $^{130}\delta_E = 1 \cdot 10^{-9} \text{ a.u.}$ energy threshold, without ($\mathbf{W} = \mathbf{A}$, left) and with spin-orbit part ($\mathbf{W} = \mathbf{A} + \mathbf{B}$, left).

is written in XFig's “Fig Format” as set of 256 gray-scale colors and, for $\mathbf{M} \in \mathbb{C}^{n \times m}$, an $n \times m$ array of gray-scale-colored square boxes. A given boxes' gray-scale color is defined by partitioning the interval

$$(0, \max_{pq} \{\log |M_{pq}|\}] \quad (5-11)$$

in 256 sub-intervals l_i of equal length, and assigning the i -th gray-scale color to the M_{pq} box if $\log |M_{pq}| \in l_i$, with l_0 being white.ⁱ Whereas this visualization is somewhat of an oversimplification from dropping M_{pq} phase (and thus sign) information, it proved to be sufficient for the purpose of this work and is, moreover, readily generalized using the same interface.

ⁱ More precisely, to prevent a large fraction of the gray-scale colors being assigned to almost-zero M_{pq} boxes, color assignment is defined by partitioning the closed interval $[t, \max_{pq} \{\log |M_{pq}|\}]$, $t > 0$; typically, $t = 1 \cdot 10^{-8}$.

6

2-Component Pseudopotentials

From the point of view of implementation, the spin-free and spin-orbit parts \hat{A} and \hat{B} of the general 2-component semilocal atomic pseudopotential

$$\begin{aligned}\hat{W} = & \underbrace{\hat{W}_{LJ} + \sum_{l=0}^{L-1} \sum_{m_l} \hat{A}_l |lm_l\rangle\langle lm_l|}_{= \hat{A}} + \underbrace{\sum_{l=0}^{L-1} \sum_{m_l} \hat{B}_l \hat{\mathbf{S}} \cdot \hat{\mathbf{L}} |lm_l\rangle\langle lm_l|}_{= \hat{B}} \quad (6-1)\end{aligned}$$

require different program functionalities from their different properties:^a In terms of matrix representations $\mathbf{W} \in \mathbb{C}^{2n \times 2n}$, \mathbf{A} decouples as $\mathbf{A}_{\alpha\alpha} \oplus \mathbf{A}_{\beta\beta}$. Moreover, from time reversal-invariance $\mathbf{A}_{\alpha\alpha} = \mathbf{A}_{\beta\beta} \in \mathbb{R}^{n \times n}$, such that spin-free PPs are readily introduced in a spin-free 1-component HF SCF framework. Contrasting, \mathbf{B} breaks spin symmetry and is complex-valued. In fact, all new classes introduced to, and all modifications of the 2006 QOL implementation's matrix and matrix representation algebra modules discussed in the preceding chapter, have been motivated solely by the consideration of non-vanishing \hat{B} operators.

However, both spin-free and spin-orbit PPs have been implemented on a common, coherent footing, i.e. employing the same class structure, PP parameter def-

^a The simpler expression of eq. 6-1 is equivalent to eq. 2-48 because, in eq. 2-48, the projection operator $\sum_{m'_l} |lm'_l\rangle\langle lm'_l|$ commutes with $\hat{\mathbf{S}} \cdot \hat{\mathbf{L}}$ and is idempotent.²⁵ Consequently, it is eq. 6-1 that is departed from for the evaluation of PP integrals over CGTO basis functions.^{31,32}

inition, parsing, and integration and integral communication techniques:

The QOL PP modules have been implemented guided by the general “pseudomolecule” design principles briefly discussed in sec. 6.1 The PP classes have been designed similarly to the QOL CGTO basis set classes and employ similar, Bison-generated parsers to process XML PP parameter definitions. PP integrals over CGTOs are evaluated using the interfaced ARGOS PP integral subroutines of Pitzer and Winter,^{31,32} and communicated as quaternion-valued integrals.

6.1 General Design Principles

Central to the design of the QOL PP modules is, first, the attempt to provide an intuitive, logically consistent class structure in line with object-oriented programming paradigms and, second, to integrate the new PP components with a minimum of changes to the interfaces of the already established 2006 QOL implementation’s modules.

The particular design – and, thus, the implementation – of the QOL PP classes has been guided by the formal analogy of the closed-shell all-electron and the valence-only PP Fock operator \hat{f} and \hat{f}_v of eq. 3-11, respectively, i.e.

$$\hat{f} = \hat{T} + \hat{U} + \hat{J} - \hat{K} \quad \text{and} \quad \hat{f}_v = \hat{T} + \hat{V} + \hat{W} + \hat{J}_v - \hat{K}_v :$$

The closed shell all-electron Fock operator \hat{f} provides, for one electron, mean fields of $2N - 1$ electrons and the nuclear Coulomb potentials U_A of all atomic nuclei A . Contrasting, the valence-only pseudopotential Fock operator \hat{f}_v sets up such fields for $2N_v - 1 < 2N - 1$ electrons only, and nuclear core-Coulomb and pseudo-potentials V_A and W_A of all atomic nuclei A . Clearly, this does not only introduce a new, i.e. the PP part \hat{W} . Instead, as compared to \hat{f} , \hat{f}_v also implies modification of global system properties as, e.g., the number of electrons and the classical Born–Oppenheimer nuclear repulsion energy.

The idea is to put eq. 3-11 at the beginning, and define a system of “pseudo-atoms” A of nuclear charge $Z_A - N_A$, each of which provides a valence–core pseudopotential $V_A + W_A$ for the system’s $2N_v = -e + \sum_A (Z_A - N_A)$ electrons, if e is the total system charge.

In other words, the atomic nucleus label A is understood to uniquely define a parameter tuple

$$\text{Par } A := (N_A, \mathcal{A}_A(N_A)) = (N_A, \{C_{ljkA}, n_{ljkA}, \beta_{ljkA}\}_{ljk}), \quad (6-2)$$

with $\mathcal{A}_A(0) := \{0\}$, for every pseudo-atom A . $\text{Par } A$ defines the A -th valence–core pseudo-potential $\hat{U}_A = \hat{U}_{\text{Par } A} = \hat{V}_{\text{Par } A} + \hat{W}_{\text{Par } A}$ such that

$$\begin{aligned} \hat{V}_{\text{Par } A} &= -(Z_A - N_A)|\hat{\mathbf{r}} - \hat{\mathbf{r}}_A|^{-1}, \\ \hat{W}_{\text{Par } A} &= \hat{W}_{LJ} + \sum_{ljk} C_{ljkA} |\hat{\mathbf{r}} - \hat{\mathbf{r}}_A|^{n_{ljkA}} e^{-\beta_{ljkA}|\hat{\mathbf{r}} - \hat{\mathbf{r}}_A|^2} \sum_{m_j} |ljm_j\rangle\langle ljm_j|, \end{aligned}$$

where an analog expansion of the local part \hat{W}_{LJ} has been omitted for clarity.

A closed shell “pseudo-molecule” of such pseudo-atoms A has $2N_v$ electrons and a nuclear repulsion energy $E_{\text{BO}}(\{Z_A - N_A\}_A)$ without any need to change the meaning of the respective attributes; only the “atom” concept employed is slightly different. Moreover, the pseudo-molecule concept includes the all-electron as special case $\sum_A N_A = 0$, recovering $\sum_A \hat{U}_{\text{Par } A} = -\sum_A Z_A |\hat{\mathbf{r}} - \hat{\mathbf{r}}_A|^{-1}$ directly from $\mathcal{A}_A(0) := \{0\}$.

Guided by these general design principles, and as discussed in the next section, every `QOL::Molecule::Atom` is assigned a PP – which is, however, a null PP in most cases.^b For the ease of notation, the pseudo-molecule concept will also be

^b An alternative design and implementation strategy is to put the molecular system of interest at the beginning, and refer to eq. 3-11 as the definition of a mean-field approximation to the solution of a (non-relativistic or approximate relativistic) $2N_v$ -electron model Schrödinger equation. In place of the established top-level `StandardMolecularHamiltonianRepresentation`, a somewhat analog “`StandardMolecularModelHamiltonianRepresentation`” class could then provide an Atom-to-PP assignment and assemble the `V` and `W` matrices similarly to `U`.

This alternative strategy is closer to eq. 3-11 on a conceptual level, i.e. regarding its formal placement among *ab initio* electronic structure theories. However, this was at the expense of having to move `Molecule` methods as `nElectrons()` and `nuclearPotential()` to the model Hamiltonian class, and having to interrelate `V` and `W` matrix assembly, also for the all-electron case. The pseudo-molecule concept, on the other side, allows to restrict all changes of the 2006 QOL implementation to a small number of class definitions – not interfaces! – in the `QOL::Molecule` namespace, and to leave the implementation of intuitively understood concepts as, e.g., the number of electrons, unchanged.

employed on the level of presentation, i.e. the valence-only index of, e.g., \hat{f}_v , $2N_v$ etc. will be discarded, and both pseudo-molecules and -atoms will be referred to as “molecules” and “atoms”.

6.2 Pseudopotential Definition and Classes

In the implementation of discussed general design principles discussed in sec. 6.1, the 2006 QOL implementation’s `QOL::Molecule::Atom` class has been given a `SemilocalAtomicPseudopotential` attribute^c as presented in sec.s 6.2.2 and 6.2.3. As construction of a particular `SemilocalAtomicPseudopotential` object is thus interrelated with the construction of `Atom` and `Molecule`, the following discussion is organized “sequentially”, i.e. following the line of PP parameter definition; `Atom` and `Molecule` construction and PP parameter parsing; and, finally, non-null PP definition.

6.2.1 Input and Parameter Definition

From the intimate connection of molecular, i.e.pseudo-molecular, and PP parameter information established in sec. 6.1, the PP definitions for a particular HF SCF calculation are given together with the molecular composition and structure definitions. For the simple TIH example, the molecular input file then reads

```
geometry = {
    Tl    0.0          0.0 0.0
    H     3.533787599 0.0 0.0
};

pseudopotential = {
    Tl "pseudopotentials/Tl_ECP60MDF/Tl_ECP60MDF"
};
```

The hydrogen null PP is implied by the missing PP parameter file path statement for H. Consequently, the complete `pseudopotential` token is, from the point of view of input processing, completely optional and recovers the 2006 QOL implementation’s use-case if absent.

^c More precisely, this attribute is of a `boost::shared_ptr` of `SemilocalAtomicPseudopotential` type.

6: 2-Component Pseudopotentials

```
<AtomicPseudopotentialEntry>
  <Comment> -----
    B. METZ, M. SCHWEIZER, H. STOLL, M. DOLG, W. LIU: THEOR. CHEM. ACC. 104, 22 (2000)
  -----</Comment>
<CoreSize> 60 </CoreSize>
<PseudopotentialAngularMomentumGroup l="0">
  <Exponent> 12.167805 <Coefficient> 281.284663 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 8.294909 <Coefficient> 62.434251 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
<PseudopotentialAngularMomentumGroup l="1">
  <Exponent> 9.891073 <Coefficient> 72.299253 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 9.003391 <Coefficient> 144.558037 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 7.151492 <Coefficient> 4.633408 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 5.172865 <Coefficient> 9.341756 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
<PseudopotentialAngularMomentumGroup l="2">
  <Exponent> 7.130218 <Coefficient> 35.943039 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 6.926904 <Coefficient> 53.909593 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 5.417570 <Coefficient> 10.381939 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 5.138681 <Coefficient> 15.583822 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
<PseudopotentialAngularMomentumGroup l="3">
  <Exponent> 5.626394 <Coefficient> 15.825488 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 5.548952 <Coefficient> 21.104021 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 2.874946 <Coefficient> 2.915127 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 2.821451 <Coefficient> 3.896903 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
<PseudopotentialAngularMomentumGroup l="4">
  <Exponent> 6.679057 <Coefficient> -7.494534 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 6.706835 <Coefficient> -9.540575 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 7.209284 <Coefficient> -7.797992 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 7.070964 <Coefficient> -9.259524 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
<PseudopotentialAngularMomentumGroup l="L">
  <Exponent> 1 <Coefficient> 0 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
<PseudopotentialAngularMomentumGroup l="1">
  <Exponent> 9.891072 <Coefficient> -144.598506 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 9.003391 <Coefficient> 144.558037 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 7.151492 <Coefficient> -9.266817 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 5.172865 <Coefficient> 9.341756 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
<PseudopotentialAngularMomentumGroup l="2">
  <Exponent> 7.151492 <Coefficient> -35.943039 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 5.172865 <Coefficient> 35.939729 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 9.891072 <Coefficient> -10.381939 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 9.003391 <Coefficient> 10.389215 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
<PseudopotentialAngularMomentumGroup l="3">
  <Exponent> 5.626394 <Coefficient> -10.550326 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 5.548952 <Coefficient> 10.552010 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 2.874946 <Coefficient> -1.943418 </Coefficient> <n> 2 </n> </Exponent>
  <Exponent> 2.821451 <Coefficient> 1.948451 </Coefficient> <n> 2 </n> </Exponent>
</PseudopotentialAngularMomentumGroup>
</AtomicPseudopotentialEntry>
```

Fig. 6.1: QOL pseudopotential parameter definition for the g_1 Tl small-core MC-DHF PP of Metz et al.¹³¹ Parameters have been sorted by increasing k and, within, increasing $j = l \pm \frac{1}{2}$. The $l = 4$ spin-orbit part has been discarded because of the limitation of the ARGOS PP integral subroutines to $l \leq 3$ for the \hat{B} part.

Non-empty parameter file path statements are processed relative to the \$HOME/QOLbasis2/ directory and define locations of XML data files, the “.xml” suffix

understood. The given example defines the $_{81}\text{Tl}$ small-core MC-DHF PP of Metz et al.,¹³¹ located at \$HOME/QOLBasis2/pseudopotentials/Tl_ECP60MDF/; the particular PP parameter definition is showcased in fig. 6.1.

Generally, PP parameters are organized for spin-free and spin-orbit PP parts separately.^d Within both parts, parameters are organized Gaussian exponent-wise, i.e. every β_{ljk} is associated a X_{ljk} and n_{ljk} wrapped in Coefficient and n tags, respectively. All such Exponent units with common l are grouped in PseudopotentialAngularMomentumGroup tags that have a single, l -valued orbital angular momentum quantum number attribute l. The PseudopotentialAngularMomentumGroup applies to both the PP \hat{A} and \hat{B} parts; these are separated by the local, l="L" part that is, thus, mandatory. PP core size and optional commentary statements are given within separate CoreSize and Comment tags, respectively.

6.2.2 Modifications to the Atom and Molecule Classes: Parsing

Instead of being supplemented by an “owned” SemilocalAtomicPseudopotential, the QOL Atom class has been given a boost::shared_ptr to SemilocalAtomicPseudopotential attribute. This is connected to the fact that, at the time of the construction of Atom from the molecular composition and structure data given in the input **geometry** token, no PP parameter definitions are available. In fact, these become available not before entering the **pseudopotential** token, i.e. not before having finished the construction of all Atom objects.

Therefore, every Atom’s shared_ptr to SemilocalAtomicPseudopotential is set to point to a null PP upon construction – which is the default setting for the all-electron case and, generally, for most non-heavy atoms –, and re-set to point to a non-null PP defined at a later time via the void define_SemilocalAtomicPseudopotential(**const** string & SP) method.

^d This seems disadvantageous compared to the more compact parametrization of eq. 2-45. However, the PP parameter definition as showcased in fig. 6.1 is closer to the actual implementation of the QOL PP classes, provides intuitive separation of PP \hat{A} and \hat{B} parts, and allows for easy internal consistency check.

```
class Atom {
public:

    Atom(const Point3D<double> & P, const TypeClass & TC,
        string Label = string()) :
        _p(P), _tc(TC), _label(Label), _sptrPP() {
        _sptrPP =
            shared_ptr<SemilocalAtomicPseudopotential<double>>(new
                SemilocalAtomicPseudopotential<double>(TC, P));
    }

    void
    define_SemilocalAtomicPseudopotential(const string & SP) {
        _sptrPP =
            shared_ptr<SemilocalAtomicPseudopotential<double>>(new
                SemilocalAtomicPseudopotential<double>(SP, _tc, _p));
    }

    // ...
private:

    Point3D<double> _p;
    TypeClass _tc;
    string _label;
    shared_ptr<SemilocalAtomicPseudopotential<double>> _sptrPP;
}
```

In this way every `SemilocalAtomicPseudopotential` object is defined only once, albeit not at the same time as `Atom`.

Note that the class interface, i.e. the parameter list of the constructor defined in lines 4–10, remains essentially unchanged.

The actual PP parameter definitions become available in the course of reading the `pseudopotential` token after construction of `Molecule` from all `Atom` objects defined in `geometry`. `Molecule` is a `std::vector<Atom>` by inheritance and has therefore – with the exceptions of the `nElectrons()` and `NuclearPotential()` member functions – not been modified directly.

However, the `Molecule` constructor from `istream` & employs a specialized `Parser` class to read molecular composition and structure definition files that, as

discussed in sec. 6.2.1, include the PP definitions as pathes to XML PP parameter files. Consequently, the Parser implementation had to be modified to allow reading of the **pseudopotential** token's contents:^e

pseudopotential holds a (possibly empty) list of pairs of element symbols and PP parameter file pathes. If Parser finds a non-empty PP definition, Parser iterates over all of Molecule's Atom entries and, if the element symbol matches the current Atom's TypeClass attribute, calls that Atom's define_SemilocalAtomicPseudopotential with the PP parameter file path as argument. As can be inferred from lines 14–16, and as discussed in sec. 6.2.3, this constructs a new SemilocalAtomicPseudopotential from the PP parameter file and the current Atom's _tc and _p attributes, corresponding to the atomic element symbol and position \mathbf{r}_A , respectively, and re-sets _spptrPP.

As already briefly stated, the only direct modifications of Molecule concern the implementation of the nElectrons() and nuclearPotential() methods to return $\sum_A (Z_A - N_A)$ and

$$E_{\text{BO}}((Z_A - N_A), \mathbf{r}_A)_A = \sum_{A > B} \frac{(Z_A - N_A)(Z_B - N_B)}{|\mathbf{r}_A - \mathbf{r}_B|},$$

respectively.

6.2.3 QOL 2-Component Semilocal Atomic Pseudopotentials

The QOL PP classes have been designed to provide a single, general class structure applying to both spin-free 1-component and spin-orbit 2-component HF SCF calculation schemes. By eq.s 2-49 and 2-50 both the PP \hat{A} and \hat{B} part's radial potential functions $X_l \in \{A_l, B_l\}$ can be written in the form as, discarding the

^e Put somewhat simplified, the 2006 QOL implementation's Parser class is a GNU Bison 2.3-generated LALR(1) parser that constructs a `std::vector` of `QOL::Molecule::Atom` from which `Molecule` is derived. The rules defining Atom assembly from the `geometry` token's contents are given by a finite-state machine-type parser table, generated by the Bison parser generator from a Backus–Naur form context-free grammar. This grammar has been supplemented to also enable reading – and, along these lines, check for syntactic correctness – of the **pseudopotential** token.

atomic core label A and implying $\mathbf{r}_A = \mathbf{0}$ for clarity,

$$X_l(\mathbf{r}) = \sum_{jk} X_{ljk} \cdot |\mathbf{r}|^{n_{ljk}} e^{-\beta_{ljk}|\mathbf{r}|^2} \quad (6-3)$$

where the expansion coefficients $X_{ljk} \in \{A_{ljk}, B_{ljk}\}$ are related to the parameters $\{C_{ljk}\}_{ljk}$ of eq. 2-45 via

$$(2l + 1)A_{ll\pm k} = v_{l\pm} C_{ll\pm k}, \quad (6-4)$$

$$(2l + 1)B_{ll\pm k} = \pm C_{ll\pm k} \quad \text{for } l \geq 1 \quad (6-5)$$

with the short-hand notations $l^\pm := l \pm \frac{1}{2}$ and $v_{l\pm} := l^\pm + \frac{1}{2}$.

From the central role of the radial potential function expansion of eq. 6-3, two auxiliary classes `PP_GaussianExponentCoefficientPair` and `PP_AngularMomentumGroup` have been implemented as template classes of a single, numerical precision-defining argument `T`. `PP_GaussianExponentCoefficientPair` is a `std::pair<T, T>` by inheritance, modeling the (β_{ljk}, X_{ljk}) pair of each term of eq. 6-3; the single attribute `_n` defines this term's radial exponent n_{ljk} . `PP_AngularMomentumGroup` has been derived from a `std::vector` of `PP_GaussianExponentCoefficientPair` entries, and models the whole expansion of X_l . Two additional `bool` attributes define locality and spin-dependency to discriminate otherwise analog expressions for \hat{W}_{LJ} , \hat{A} , and \hat{B} ; the third, `unsigned int` attribute `_l` defines `PP_AngularMomentumGroup`'s l value.

The top-level single-parameter `SemilocalAtomicPseudopotential` template class of argument `T` has, then, been derived from a `std::vector` of `PP_AngularMomentumGroup`. To account for the full complexity of the general expression of eq. 2-45, `SemilocalAtomicPseudopotential` has had to be given a number of atomic core-related `TypeClass`-, `unsigned int`-, and `Point3D`-type attributes defining the atomic core's element symbol, N_A , and \mathbf{r}_A , respectively, and two more `unsigned int` attributes defining the PP \hat{A} and \hat{B} part's expansion lengths $L - 1$.^f

^f The obvious implementation alternative of employing, e.g., a pointer or reference to the associated `Atom` object is prohibited as `Atom` itself has a `shared_ptr` of `SemilocalAtomicPseudopotential`, which would result in a cyclic dependence of these. However, `SemilocalAtomicPseudopotential` must carry the atomic core information as these are to be passed to the ARGOS PP

Construction of `SemilocalAtomicPseudopotential` from `const string &`, `const TypeClass &`, and `Point3D<double>` by Atom's `define_SemilocalAtomicPseudopotential` method, alongside PP parameter parsing, involves the 2006 QOL implementation's XML-parsing modules, particularly the `QOL::XML::XMLLoader` and `QOL::XML::XMLTreeInputIterator`, in a largely analog fashion as employed in the construction of the CGTO basis set classes, and will not be discussed in any detail.

6.3 Pseudopotential Integrals

QOL PP matrix elements between CGTO basis functions are evaluated using parts of the ARGOS program^{31–35} of Pitzer et al.^g

Guided by the discussion of sec. 4.3.2, the interface of the ARGOS PP integral subroutines has been designed to

- evaluate PP matrix elements $W_{pq\gamma\gamma'}$ between primitive non-normalized CGTO basis functions only; and
- to evaluate $W_{pq\gamma\gamma'}$ atom-wise and sum over all atomic PP contributions of the molecular PP afterwards.

Whereas the first point is motivated by interoperability reasons – and effectively allows to employ of the 2006 QOL implementation's iterator–evaluator structure –, the second allows to circumvent, to some extent, ARGOS array dimension limitations, and eases ARGOS subroutine and QOL module communication. Clearly, both spoil efficient construction of the `W` matrix, but this is considered insignificant as the computational effort of any HF SCF calculation is exclusively determined by 4-index integral evaluation and/or (repeated) $\mathbf{J} - \mathbf{K}$ matrix assembly.

The atom-wise evaluation of PP matrix elements allows to employ the lower-level ARGOS subroutines `PSEUD1`, `PSEUD2`, and `PSEUD3` briefly introduced in sec.

integral subroutines alongside the PP parameters, but without explicit reference to the particular `Atom` object.

^g The “Argonne, Ohio State” program ARGOS is a general-purpose integral program for the evaluation of S_{pq} , T_{pq} , U_{pq} , A_{pq} , B_{pq} , g_{prqs} , and a number of property integrals over symmetry-adapted, generally contracted CGTO functions, from the COLUMBUS suite of programs.^{132–135}

6.3.1. Juxtaposing these subroutines' interfaces with the 2006 QOL implementation's iterator–evaluator structure, discussed in sec. 4.3.2, almost directly defines the QOL PP integral evaluator classes and QOL PP integral communication, as presented in sec.s 6.3.2 and 6.3.3.

6.3.1 ARGOS Integral Subroutines: Overview

ARGOS spin-free and spin–orbit PP integrals over symmetry-adapted generally contracted CGTO basis functions are evaluated within the McMurchie–Davidson recursion scheme:^{31,32h}

Matrix elements of the local \hat{W}_{LJ} , the non-local spin-free $\hat{A} - \hat{W}_{LJ}$, and the non-local spin–orbit \hat{B} part of the general 2-component semilocal atomic pseudopotential operator of eq. 6-1 are referred to as type 1, type 2, and type 3 integrals, and computed by PSEUD1, PSEUD2, and PSEUD3, respectively. All are FORTRAN-written, closely related subroutines that are called with analog sets of six parameters, e.g., for the \hat{W}_{LJ} part,

SUBROUTINE PSEUD1(CCR, GOUT, NCR, NKCRL, NKCRU, ZCR)

- The **REAL*8** array variables CCR and ZCR, and the **INTEGER** array variable NCR each define a maximum of 77 PP parameters X_{ljkA} , β_{ljkA} , and n_{ljkA} , respectively; the spin–orbit part's expansion coefficients $X_{ljkA} = B_{ljkA}$ are required to be divided by $2l + 1$.
- GOUT is a **REAL*8** array variable that, on exit, contains the integrals over all pairs of CGTOs with the input basis functions' l_p and l_q :ⁱ As there are $\frac{1}{2}(l+1)(l+2)$ Cartesian monomials $x^{m_x} y^{m_y} z^{m_z}$ with $m_x + m_y + m_z = l$, GOUT holds

$$\frac{1}{2}(l_p + 1)(l_p + 2) \cdot \frac{1}{2}(l_q + 1)(l_q + 2)$$

^h PP integrals are first separated in angular and radial parts. Whereas angular integral evaluation is relatively straightforward, radial integration is more difficult from the integrands' modified spherical Bessel functions and involves different, (for reasons of numerical stability) case-dependent single and double power series expansion, Gaussian quadrature, and techniques of expansion in terms of scaled modified spherical Bessel functions.^{31,32}

ⁱ Note that the CGTO information is communicated over the ONE and CALLIN common blocks, as discussed below.

integrals for input CGTOs $|\phi_p\rangle$ and $|\phi_q\rangle$ with l_p and l_q .

From the ARGOS limitation to $l \leq 4$ CGTOs, GOUT's dimension is 255 for PSEUD1 and PSEUD2. In the special PSEUD3 case, GOUT is of size $3 \cdot 255 = 675$, because it holds three real integrals

$$\begin{aligned}\text{Im } B_{pq\alpha\beta} &= \sum_{l=1}^L \sum_{m_l} \langle \phi_p | \hat{B}_l \hat{L}_x | lm_l \rangle \langle lm_l | \phi_q \rangle, \\ \text{Re } B_{pq\alpha\beta} &= -i \sum_{l=1}^L \sum_{m_l} \langle \phi_p | \hat{B}_l \hat{L}_y | lm_l \rangle \langle lm_l | \phi_q \rangle, \\ \text{Im } B_{pq\alpha\alpha} &= \sum_{l=1}^L \sum_{m_l} \langle \phi_p | \hat{B}_l \hat{L}_z | lm_l \rangle \langle lm_l | \phi_q \rangle\end{aligned}$$

at GOUT positions $3i_{pq}$, $3i_{pq} + 1$, and $3i_{pq} + 2$, respectively, with respect to the position i_{pq} of a particular $(\mathbf{m}_p, \mathbf{m}_q)$ integral in the spin-free PSEUD1 and PSEUD2 cases.^j

- The **INTEGER** array variables NKTRL and NKCRU – in the special PSEUD3 case: NKSL and NKSU – of dimension 6 assist ARGOS PP parameter definition: If all PP parameter triples $(X_{ljkA}, \beta_{ljkA}, n_{ljkA})$ are listed with one such triple in each line, starting with the \hat{W}_{LJ} expansion, and being ordered by increasing l , NKTRL and NKCRU contain the numbers of the first and last lines, respectively, of all l blocks.^k

Additionally, including the auxiliary ANG1, ANG2, BESS, CINTS, COLIB1, CORTAB, FACAB, PTWT, QCMP, QPASY, and RECUR1 subroutines called by PSEUD1, PSEUD2, and PSEUD3, a larger number of variables communicated via the common blocks UNITS, PARMR, PARMI, ONE, CALLIN, QSTORE, LTAB, ZLMTAB, DFAC, PIFAC, and FACT, and 4 global **LOGICAL** variables ESF, ESFC, IGUEQ1, and JGUEQ1 are required. These include, for example, the $|\phi_p\rangle$ and $|\phi_q\rangle$ parameters needed for $W_{pq\gamma\gamma'}$ evaluation. No comprehensive discussion of the ARGOS data

^j Note that $\text{Re } \mathbf{B}_{\alpha\alpha} = \mathbf{0}_n$. For PSEUD1 and PSEUD2, a particular $(\mathbf{m}_p, \mathbf{m}_q)$ integral's GOUT position i_{pq} is defined by ARGOS' ordering of the Cartesian monomials \mathbf{m} , as collected in tab. 6.1: If \mathbf{m}_p and \mathbf{m}_q are the $\pi_{\mathbf{m}_p}$ -th and $\pi_{\mathbf{m}_q}$ -th Cartesian monomials, respectively, GOUT holds the respective integral at position

$$i_{pq} = \frac{1}{2} \pi_{\mathbf{m}_p} ((l_p + 1)(l_p + 2) + 2\pi_{\mathbf{m}_q}).$$

^k For the ^{81}Tl small-core MC-DHF PP given in fig. 6.1, NKTRL and nkcru contain the numbers 1, 2, 4, 8, 12, 16; and 1, 3, 7, 11, 15, 19, respectively. Consequently, NKSL and NKSU contain 20, 24, 28, 0, 0, 0; and 23, 27, 31, 0, 0, 0, respectively.

Tab. 6.1: Ordering of Cartesian monomials $\mathbf{m} = (m_x, m_y, m_z)$ by ARGOS: For all CGTO orbital angular momentum numbers $l \leq 4$ supported, the Cartesian monomial's position $\pi_{\mathbf{m}}$ is given, ommiting the trivial $l = 0$ case.

$\ \mathbf{m}\ _1 = l = 1$	$\ \mathbf{m}\ _1 = l = 2$	$\ \mathbf{m}\ _1 = l = 3$	$\ \mathbf{m}\ _1 = l = 4$
$\mathbf{m} \mapsto \pi_{\mathbf{m}}$	$\mathbf{m} \mapsto \pi_{\mathbf{m}}$	$\mathbf{m} \mapsto \pi_{\mathbf{m}}$	$\mathbf{m} \mapsto \pi_{\mathbf{m}}$
$(1, 0, 0) \mapsto 0$	$(2, 0, 0) \mapsto 0$	$(3, 0, 0) \mapsto 0$	$(4, 0, 0) \mapsto 0$
$(0, 1, 0) \mapsto 1$	$(0, 2, 0) \mapsto 1$	$(0, 3, 0) \mapsto 1$	$(0, 4, 0) \mapsto 1$
$(0, 0, 1) \mapsto 2$	$(0, 0, 2) \mapsto 2$	$(0, 0, 3) \mapsto 2$	$(0, 0, 4) \mapsto 2$
	$(1, 1, 0) \mapsto 3$	$(2, 1, 0) \mapsto 3$	$(3, 1, 0) \mapsto 3$
	$(1, 0, 1) \mapsto 4$	$(2, 0, 1) \mapsto 4$	$(3, 0, 1) \mapsto 4$
	$(0, 1, 1) \mapsto 5$	$(1, 2, 0) \mapsto 5$	$(1, 3, 0) \mapsto 5$
		$(0, 2, 1) \mapsto 6$	$(0, 3, 1) \mapsto 6$
		$(1, 0, 2) \mapsto 7$	$(1, 0, 3) \mapsto 7$
		$(0, 1, 2) \mapsto 8$	$(0, 1, 3) \mapsto 8$
		$(1, 1, 1) \mapsto 9$	$(2, 2, 0) \mapsto 9$
			$(2, 0, 2) \mapsto 10$
			$(0, 2, 2) \mapsto 11$
			$(2, 1, 1) \mapsto 12$
			$(1, 2, 1) \mapsto 13$
			$(1, 1, 2) \mapsto 14$

flow and variable interdependence will be given. Instead, the following presentation of the ARGOS–QOL interface is focussed on the comparably few cases that have to be addressed directly on the QOL side of the ARGOS–QOL interface.

6.3.2 ARGOS–QOL Interfacing

From the preceding discussion, calling the PSEUD1, PSEUD2, and PSEUD3 subroutines requires, first, assembly and communication of the parameter list – including re-formatting from QOL to generally different ARGOS data organization schemes – and, second, global and case-dependent common block variable setting:

The first communication level, being considered only with PP parameters and

integral array memory allocation, is enabled by a simple, single `QOL2ARGOS_PP-MapperContainer` class. The second level requires two FORTRAN-written subroutines `INIT` and `EDIT`, wrapped by the `ARGOSCommonBlockWrapper` class. PP matrix elements between CGTOs of particular Cartesian monomials in the `GOUT` arrays are located by the auxiliary `ARGOS2QOL_Cartesian-FunctionIntegral-Mapper` class.

QOL-to-ARGOS Pseudopotential Parameter Communication

To enable safe, wrapped PP parameter re-formatting to ARGOS-read array variables, an auxiliary one-parameter template class `QOL2ARGOS_PPMapperContainer` of argument `T` is provided: It has, outside a pointer to the `SemilocalAtomicPseudopotential` it refers to, flat C-array attributes `_ccr` and `_zcr` of `T`, and `_ncr`, `_nkcr1`, `_nkcr2`, `_nkls1`, and `_nkls2` of `int` types. These resemble the `CCR`, `ZCR`, `NCR`, `NKCR1`, `NKCR2`, `NKLS1`, and `NKLS2` array variables handed over to the `PSEUD1`, `PSEUD2`, and `PSEUD3` subroutines and are, accordingly, filled as discussed in sec. 6.3.1 upon construction of `QOL2ARGOS_PPMapperContainer` from `const SemilocalAtomicPseudopotential<double> &`.¹

ARGOS Common Block and PP Integral Subroutine Wrapping

As C++ does not support declaration and definition of global variables in common blocks, ARGOS common block variables are set by two FORTRAN-written subroutines `INIT` and `EDIT`:

The `INIT` subroutine is intended to be called only once for each program execution, and provides straightforward declaration and definition of all DATA statements, and UNITS, PARMR, PARMI, ONE, CALLIN, QSTORE, LTAB, ZLMTAB, DFAC, PIFAC, and FACT common blocks. Note that `INIT` does not, and cannot provide a full emulation of an ARGOS or even COLUMBUS run on the level of common

¹ Technically, both `QOL2ARGOS_PPMapperContainer` and `SemilocalAtomicPseudopotential` are templates of a numerical precision-defining argument `T`. However, as `Atom` and `Molecule` use `double` types only, typically only instances with `T` set to `double` are employed. Moreover, ARGOS employs double precision floating point, i.e. `REAL*8` types only, and FORTRAN subroutines do not allow overloading.

block variable settings, but, instead, addresses only those variables that turned out to be relevant for PP integral evaluation.

Contrasting, EDIT has been implemented to be called every time a particular PP matrix element – actually the pair $(W_{pq\alpha\alpha}, W_{pq\alpha\beta})$ – is to be evaluated and, thereby, to re-set the case-dependent variables defining PP and CGTO parameters. EDIT is, therefore, called with a total of 20 parameters collected in tab. 6.2. From these, all case-dependent ONE and CALLIN variables, and the $|\phi_p\rangle$ and $|\phi_q\rangle$ normalization constants N_p and N_q for ARGOS normalization to

$$\|\phi_p\| \mapsto \sqrt{\text{cmf } \mathbf{m}_p} := \sqrt{(2m_{px} - 1)!! \cdot (2m_{py} - 1)!! \cdot (2m_{pz} - 1)!!} \quad (6-6)$$

are calculated; eq. 6-6 defines the “Cartesian monomial double factorial” function for the ease of further discussion.

From the C++ side, two **void**-returning global functions `QOL::Pseudopotential::init` and `QOL::Pseudopotential::edit` have been defined to call the respective, **extern "C"**-declared FORTRAN subroutines `INIT` and `EDIT` with parameter lists of pointers to the respective parameters’ types.^m

`init` and `edit` are wrapped by the auxiliary `ARGOSCommonBlockWrapper` class, being a single-parameter template class of argument `T`, that provides two methods `void init_ARGOSCommonBlock()` and

```

1 void edit_ARGOSCommonBlock(
2     const CartesianGaussianFunction<T> &,
3     const CartesianGaussianFunction<T> &,
4     const SemilocalAtomicPseudopotential<double> &,
5     const bool &) const;
```

that effectively call `INIT` and `EDIT` via `init` and `edit`, respectively. All EDIT parameters of tab. 6.2 are retrieved from `CartesianGaussianFunction` and `SemilocalAtomicPseudopotential` objects’ attributes; the fourth `const bool &` parameter defines spin-free and spin-orbit use-cases, i.e. PP spin-dependency, by `false` and `true`, respectively.

^m`INIT` and `EDIT` are declared with an underbar character appended via the `QOL::Pseudopotential::FortranLinkage` pre-processor directives; declaration of `FortranLinkage(INIT)` and `FortranLinkage(EDIT)` as `extern "C"` disables C++ name mangling and, thus, polymorphism and overloading.

Wrapping ARGOS common block variable setting in this way does not only restrict the global `init` and `edit` functions to be exclusively called by methods of `ARGOSCommonBlockWrapper`, but also provides, to some extent, type-safety and simplifies use-cases.

Tab. 6.2: Variable types, names, and meanings of the `EDIT` subroutine's parameter list. `EDIT` resets `ONE` and `CALLIN` common block variables for the evaluation of PP matrix elements between CGTOs $\phi_p(\mathbf{r}) = N_p \cdot x^{m_{px}} y^{m_{py}} z^{m_{pz}} \exp(-\zeta_p |\mathbf{r} - \mathbf{r}_p|^2)$, for a single atomic PP at \mathbf{r}_A , with expansion length $L - 1$.ⁿ

type	name	meaning
REAL*8	IX__, IY__, IZ__; IMX__, IMY__, IMZ__	$\mathbf{r}_p; \mathbf{m}_p$
REAL*8	JX__, JY__, JZ__; JMX__, JMY__, JMZ__	$\mathbf{r}_q; \mathbf{m}_q$
INTEGER	IL__; JL__	$l_p; l_q$
REAL*8	IA__; JA__	$\zeta_p; \zeta_q$
REAL*8	X__, Y__, Z__	\mathbf{r}_A
INTEGER	L__	L

In a similar way, the global functions `pseud1`, `pseud2`, and `pseud3`, implemented in the `QOL::Pseudopotential` namespace, call the ARGOS `PSEUD1`, `PSEUD2`, and `PSEUD3` subroutines with parameter lists of pointers to the respective parameters' types.

Locating Integrals over CGTO Pairs with Specified Cartesian Monomials

For easy, intuitive retrieval of particular integrals over CGTOs $|\phi_p\rangle$ and $|\phi_q\rangle$ with given Cartesian monomials \mathbf{m}_p and \mathbf{m}_q from the `GOUT` arrays, the auxiliary class `ARGOS2QOL_CartesianFunctionIntegralMapper` has been derived from `vector<map<QOL::AngularBasis::CartesianFunction, unsigned int>>`: This design effectively mimicks tab. 6.1, employing the vector components as angular momentum quantum number l . The only member function `read`

ⁿ The Cartesian monomials $\mathbf{m} \in \mathbb{N}^3$ are, despite their components' non-negative integer nature, represented by `REAL*8` types to take into account the full domain, i.e. \mathbb{Q} , of the double factorial function.

of ARGOS2QOL_CartesianFunctionIntegralMapper takes a **const** CartesianFunction & CF as argument and, via

```
return (*this)[CF.sum()].find(CF)->second
```

gives the number $\pi_{\mathbf{m}}$ of the input Cartesian monomial \mathbf{m} , thus locating $A_{pq\alpha\alpha}$, $\text{Im } B_{pq\alpha\beta}$, $\text{Re } B_{pq\alpha\beta}$, and $\text{Im } B_{pq\alpha\alpha}$ in GOUT as discussed in sec. 6.3.1.

6.3.3 Integration and Pseudopotential Integral Communication

Whereas the particular integral evaluation procedure is almost completely defined by the ARGOS–QOL interface’s class structure of sec. 6.3.2, the top-level evaluator class interface is less independent of the 2006 QOL implementation if the established iterator–evaluator structure is to be re-used without modifications. As discussed in sec. 4.3.2, the QOL Buffered_IteratorEvaluator1 class iterates over all symmetry-unique CGTO pairs $(|\phi_p\rangle, |\phi_q\rangle) \in I \subset B \times B$ and, of course, only once. – However, for any CGTO pair in I , both $W_{pq\alpha\alpha}$ and $W_{pq\alpha\beta}$ are to be evaluated.

Therefore, QOL PP matrix elements are communicated quaternion-valued by setting the template argument T of the evaluator class interface

```
1 // typedef typename T::value_type _Field;
2 template <class T>
3 T McMDPseudopotential_Evaluator<T>::operator () (
4     const CartesianGaussianFunction<_Field> & bra,
5     const CartesianGaussianFunction<_Field> & ket ) const;
```

to QOL::ElementaryMath::Quaternion, employing the maps \mathcal{N} and \mathcal{M} of sec. 5.1.2.

Pseudopotential Integral Evaluation Interface

QOL::Pseudopotential::McMDPseudopotential_Evaluator is a single-parameter template class of argument T, closely resembling the 2006 QOL implementation’s evaluator classes. The single constructor from **const** Molecule & provides the molecular PP information and assembles the evaluator classes’ std::vector attribute that, for every non-null atomic PP, has a std::pair of SemilocalAtomicPseudopotential and QOL2ARGOS_-PPMapperContainer entry to

assist PP parameter information communication to the underlying ARGOS PP integral subroutines. ARGOS common block variable setting and integral retrieval from GOUT is enabled by `ARGOSCommonBlockWrapper` and `ARGOS2QOL_CartesianFunctionIntegralMapper` attributes.

T is intended to be of a Quaternion-type that, through the time reversal-symmetry of the PP operators, enables to wrap both $W_{pq\alpha\alpha}$ and $W_{pq\alpha\beta}$ as a single quaternion

$$\mathcal{N}(W_{pq\alpha\alpha}, W_{pq\alpha\beta}) = A_{pq\alpha\alpha} + iB_{pq\alpha\alpha} + j\text{Re } B_{pq\alpha\beta} + k\text{Re } B_{pq\alpha\beta}$$

using the $\mathbb{C}^{n \times n} \times \mathbb{C}^{n \times n} \rightarrow \mathbb{H}^{n \times n}$ map \mathcal{N} of eq. 5-3 for the special $n = 1$ case.

As PP integrals are evaluated atom-wise, calling the parentheses operator involves, after initialization of the return quaternion `integrals`, iteration over all entries of the vector of pair of `SemilocalAtomicPseudopotential` and `QOL-2ARGOS_PPMapperContainer` attributes. For every such entry pointed at by the employed `const_iterator` A ,

- C-arrays `scratch1`, `scratch2`, and `scratch3` of sizes 255, 255, and 675 are defined. After setting the case-dependent ARGOS common block variables by the `ARGOSCommonBlockWrapper<double> _CB`'s `edit_ARGOSCommonBlock`, the wrapped ARGOS subroutines `pseud1`, `pseud2`, and `pseud3` are called to evaluate the current atomic PP's contributions to $W_{pq\alpha\alpha}$ and $W_{pq\alpha\beta}$. For example, for the spin-orbit integrals,

```

1 double scratch3[675];
2 memset(scratch3, 0.0, 675*sizeof(double));
3
4 _CB.edit_ARGOSCommonBlock(ket, bra, A->first, true);
5 QOL::Pseudopotential::pseud3(
6     A->second._ccr, scratch3,
7     A->second._ncr, A->second._nkls1,
8     A->second._nklsu, A->second._zcr
9 );

```

Note that the bra and ket CGTOs have to be interchanged with respect to the QOL index definition. Then,

- $A_{pq\alpha\alpha}$, $B_{pq\alpha\alpha}$, $\text{Re } B_{pq\alpha\beta}$, and $\text{Im } B_{pq\alpha\beta}$ corresponding to the bra and ket CGTOs' Cartesian monomials are located in the `scratch1`, `scratch2`, and

`scratch3` arrays by `ARGOS2QOL_CartesianFunctionIntegralMapper`, and added to the `i`, `j`, and `k` components of `integrals`, respectively.^o

After the loop over all atomic PP contributions has been finished, `integrals` is re-“normalized” to non-normalized CGTOs, i.e.

$$\mathcal{N}(W_{pq\alpha\alpha}, W_{pq\alpha\beta}) \mapsto \mathcal{N}(W_{pq\alpha\alpha}, W_{pq\alpha\beta}) \cdot \frac{\|\phi_p\| \cdot \|\phi_q\|}{\sqrt{\text{cmf } \mathbf{m}_p \cdot \text{cmf } \mathbf{m}_q}}, \quad (6-7)$$

because normalization – again to $\sqrt{\text{cmf } \mathbf{m}}$ – is done in the course of contraction of the primitive integrals to integrals over contracted (and possibly Cartesian-to-spherical-transformed) CGTOs by the 2006 QOL implementation’s iterator-evaluator modules.

Integral Communication and Matrix Representation Construction

From the possibility to wrap both $W_{pq\alpha\alpha}$ and $W_{pq\alpha\beta}$ in a single quaternion-valued return type, PP integral evaluation is readily incorporated in the 2006 QOL implementation’s module architecture:

The top-level class `StandardMolecularHamiltonian2SpinorRepresentation_Container` has been supplemented with the necessary attributes to allow a method `PP()` to return a `ContainerRepresentation` object with, similar to the discussion of sec. 4.3.2, third and fourth template arguments `BufferedIteratorEvaluator1` and `OneBody_Evaluator` of `McMDPseudopotential_Evaluator` of `Quaternion`, respectively.^p

On the matrix representation classes’ side, the `HermitianTimeReversalInvariantMatrix` constructor from `ContainerRepresentation` – more precisely: the template constructor from a `ContainerRepresentation`-type instance of `CR`

^o It is necessary to clean the `PSEUD1`, `PSEUD2`, and `PSEUD3` subroutines’ `GOUT` arrays from almost-zero entries before adding up `integrals`, as these entries have been observed to severely break spin symmetry in HF SCF calculations employing the PP \hat{A} part only. The present implementation sets all `scratch1`, `scratch2`, and `scratch3` entries smaller than 10^{-20} to zero before PP integrals are processed any further.

^p A very small number of modifications to the `Buffered_IteratorEvaluator1` implementation has had to be made to allow contraction of quaternion-valued integrals using the established loop structures, i.e., replacing the employed $O_{pq} = O_{qp}$ by the more general $O_{pq} = O_{qp}^*$ relation.

- has been specialized for McMDPpseudopotential_Evaluator-valued Operator arguments of the OneBody_Evaluator argument of Buffered_Iterator-Evaluator1: The implementation is closely analog to the 2006 QOL implemen-

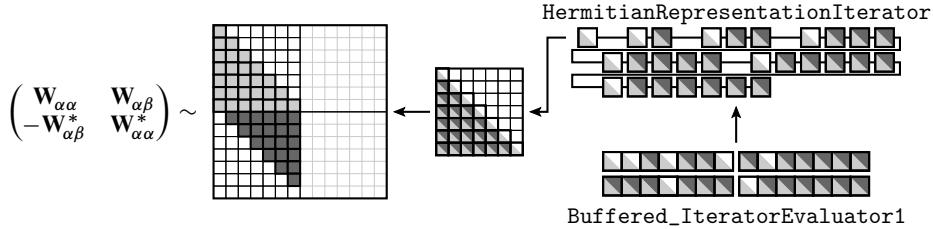


Fig. 6.2: Construction of general 2-component pseudopotential operator matrix representations: HermitianRepresentationIterator maps the $\frac{1}{2}n(n + 1)$ quaternions to a sequence $(W_{pq})_{p \geq q} \subset \mathbb{H}$ and, thus, to the lower triangular part of the hermitian matrix $\mathbf{M} \in \mathbb{H}^{n \times n}$. Then, \mathbf{M} is sent to $\mathbf{Q} \in \mathbb{A}_n \subset \mathbb{C}^{2n \times 2n}$ via \mathcal{M} .

tation's construction of spin-free operator matrix representation objects. Instead of a UnitarySpace::ScalarProduct::ScalarProduct_Codomain-valued SymmetricMatrixRepresentation, however, a Quaternion-valued HermitianMatrix is constructed in an otherwise identical fashion. This is readily mapped to a HermitianQuaternionRepresentationMatrix

$$\mathbf{Q} = (\mathcal{M} \circ \mathcal{N})(\mathbf{W}_{\alpha\alpha}, \mathbf{W}_{\alpha\beta}),$$

using the $\mathbb{H}^{n \times n} \rightarrow \mathbb{A}_n$ matrix isomorphism \mathcal{M} as illustrated in fig. 6.2, and cast to the derived HermitianTimeReversalInvariantMatrixRepresentation.

7

Self-Consistent 2-Spinor Fields

7.1 Spin Component Exchange Coupling

As an indirect consequence of the spin symmetry-breaking properties of the spin-orbit pseudopotential operator \hat{B} , the HF SCF Exchange operator couples the 2-spinor's α - and β -spin components by eq. 3-42, i.e.

$$K_{pq\gamma\gamma'} = - \sum_{rs} g_{prsq} D_{sr\gamma\gamma'}, \quad (7-1)$$

which must, thus, be implemented differently from the spin-free 1-component HF SCF calculation scheme.

To provide similar interfaces for both the 1-component and the 2-component case, two separate functions `assemble_1cSCF2ParticlePart` and `assemble_2cSCF2ParticlePart` have been implemented analog to the 2006 QOL implementation's `assembleRHF2ParticlePart2` function in the `QOL::CartesianGaussianIntegration` namespace. The following discussion is focussed exclusively on the 2-component case; specialization to the simpler 1-component case is straightforward by formally setting $K_{pq\gamma\gamma'}$ to $K_{pq\gamma\gamma'}\delta_{\gamma\gamma'} \in \mathbb{R}$ and, technically, employing the respective spin-free 1-component matrix and matrix representation classes of sec. 4.3

The arguments taken by `assemble_2cSCF2ParticlePart`, i.e.

```

1 // typedef
2 // HermitianTimeReversalInvariantMatrixRepresentation<
3 //     UnitarySpace> Mrep;
4 template <class UnitarySpace>
5 Mrep assemble_2cSCF2ParticleParts(
6     const CartesianGaussian_UnitarySpace2<typename
7         UnitarySpace:::_Basis> & BtensorB,
8     const Mrep & D,
9     const vector<typename UnitarySpace:::_Field> & G_n ) {

```

correspond to the unitary space $\mathbb{B} \otimes \mathbb{B}$ spanned by the CGTO products $|\phi_p \phi_q\rangle$, the density matrix \mathbf{D} , and the set \mathcal{G}_n of all symmetry-unique 4-index integrals g_{prqs} , respectively.

In the integral-driven assembly of the exchange operator matrix representation – as briefly discussed for the 2006 QOL implementation in sec. 4.3.3 – iteration is over all symmetry-unique 4-index integrals $g_{prqs} \in \mathcal{G}_n$. As a given integral g_{prqs} contributes to several $K_{pqyy'}$, for every such g_{prqs} all $v_{prqs} \leq 8$ unique index combinations^a are to be identified with the corresponding elements of \mathbf{D} , and with the elements of \mathbf{K} they contribute to.

This procedure is schematically illustrated by tab. 7.1: For example, if $p = q$ and $r = s$ (given as the next to last case in tab. 7.1), g_{prpr} contributes, both as g_{prpr} and g_{rprp} , to $K_{pryy'}$ and $K_{rpYY'}$, respectively; from eq. 3-42, every such 4-index integral in \mathcal{G}_n is to be multiplied by $-D_{pryy'}$ and $-D_{rpYY'}$ and added to $K_{pryy'}$ and $K_{rpYY'}$, respectively.

This straightforward assembly scheme can be significantly simplified by consid-

^a The number v_{prqs} of unique index combinations depends on the given index values, i.e. on which of six possible sets of index equalities among these values apply. From

$$\begin{aligned}
 g_{prqs} &= \langle \phi_p \phi_r | \hat{g} \phi_q \phi_s \rangle = \iint_{\mathbb{R}^3 \times \mathbb{R}^3} d\mathbf{r}_1 d\mathbf{r}_2 \phi_p^*(\mathbf{r}_1) \phi_r^*(\mathbf{r}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \phi_q(\mathbf{r}_1) \phi_s(\mathbf{r}_2) \\
 &\Rightarrow g_{prqs} = g_{rpsq} = g_{qspr}^* = g_{sqrp}^*
 \end{aligned}$$

due to electron-electron symmetry and hermiticity of \hat{g} . Moreover, since $\mathcal{G}_n \subset \mathbb{R}$ for real-valued CGTOs $|\phi_p\rangle$, one can electron-1 and -2 indices independently, i.e. $g_{prqs} = g_{rqps} = g_{psqr} = g_{qspr} = g_{rpsq} = g_{sprq} = g_{rqsp} = g_{sqrp}$, to arrive at $v_{prqs} = 8$ for pairwise distinct index values. Clearly, for any other identity relation among the index values, a smaller number of distinct 4-index integrals arises; for example, for $r = s$, $v_{prqr} = 4$.

Tab. 7.1: Construction of $\mathbf{K} \in \mathbb{C}^{2n \times 2n}$ by `assemble_2cSCF2ParticleParts`: The given symbolic expressions, e.g., “ $g_{prqs} : \times -D_{q\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$ ” are read as: “ g_{prqs} is to be multiplied by $-D_{q\gamma\gamma'}$, and this product is to be added to $K_{p\gamma\gamma'}$ ”. The \blacktriangleleft - and \triangleleft -labeled contributions to $\mathbf{K}_{\alpha\alpha}$ and $\mathbf{K}_{\alpha\beta}$ are non-redundant as discussed in the text.

	$\mathbf{K}_{\alpha\alpha} \in \mathbb{C}^{n \times n}$	$\mathbf{K}_{\alpha\beta} \in \mathbb{C}^{n \times n}$	
$= g_{prqs} : \times -D_{q\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{ps\alpha\alpha}$	$K_{ps\alpha\beta}$	◀
$= g_{qrps} : \times -D_{p\gamma\gamma'} \mapsto K_{qs\gamma\gamma'}$:	$K_{qs\alpha\alpha}$	$K_{qs\alpha\beta}$	◀
$= g_{psqr} : \times -D_{q\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{pr\alpha\alpha}$	$K_{pr\alpha\beta}$	◀
$= g_{qspr} : \times -D_{p\gamma\gamma'} \mapsto K_{q\gamma\gamma'}$:	$K_{qr\alpha\alpha}$	$K_{qr\alpha\beta}$	◀
$= g_{rpsq} : \times -D_{s\gamma\gamma'} \mapsto K_{r\gamma\gamma'}$:	$K_{rq\alpha\alpha} = K_{qr\alpha\alpha}^*$	$K_{rq\alpha\beta} = -K_{qr\alpha\beta}$	
$= g_{sprq} : \times -D_{r\gamma\gamma'} \mapsto K_{s\gamma\gamma'}$:	$K_{sq\alpha\alpha} = K_{qs\alpha\alpha}^*$	$K_{sq\alpha\beta} = -K_{qs\alpha\beta}$	
$= g_{rqsp} : \times -D_{s\gamma\gamma'} \mapsto K_{r\gamma\gamma'}$:	$K_{rp\alpha\alpha} = K_{pr\alpha\alpha}^*$	$K_{rp\alpha\beta} = -K_{pr\alpha\beta}$	
$= g_{sqrp} : \times -D_{r\gamma\gamma'} \mapsto K_{s\gamma\gamma'}$:	$K_{sp\alpha\alpha} = K_{ps\alpha\alpha}^*$	$K_{sp\alpha\beta} = -K_{ps\alpha\beta}$	
$= g_{prqr} : \times -D_{q\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{pr\alpha\alpha}$	$K_{pr\alpha\beta}$	◀
$= g_{qrpr} : \times -D_{p\gamma\gamma'} \mapsto K_{q\gamma\gamma'}$:	$K_{qr\alpha\alpha}$	$K_{qr\alpha\beta}$	◀
$= g_{rprq} : \times -D_{r\gamma\gamma'} \mapsto K_{r\gamma\gamma'}$:	$K_{rq\alpha\alpha} = K_{qr\alpha\alpha}^*$	$K_{rq\alpha\beta} = -K_{qr\alpha\beta}$	
$= g_{qrqp} : \times -D_{r\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{rp\alpha\alpha} = K_{pr\alpha\alpha}^*$	$K_{rp\alpha\beta} = -K_{pr\alpha\beta}$	
$= g_{prps} : \times -D_{p\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{ps\alpha\alpha}$	$K_{ps\alpha\beta}$	◀
$= g_{pspr} : \times -D_{p\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{pr\alpha\alpha}$	$K_{pr\alpha\beta}$	◀
$= g_{rpsp} : \times -D_{s\gamma\gamma'} \mapsto K_{r\gamma\gamma'}$:	$K_{rp\alpha\alpha} = K_{pr\alpha\alpha}^*$	$K_{rp\alpha\beta} = -K_{pr\alpha\beta}$	
$= g_{sprp} : \times -D_{r\gamma\gamma'} \mapsto K_{s\gamma\gamma'}$:	$K_{sp\alpha\alpha} = K_{ps\alpha\alpha}^*$	$K_{sp\alpha\beta} = -K_{ps\alpha\beta}$	
$= g_{ppqq} : \times -D_{q\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{pq\alpha\alpha}$	$K_{pq\alpha\beta}$	◀
$= g_{qppq} : \times -D_{p\gamma\gamma'} \mapsto K_{q\gamma\gamma'}$:	$K_{qq\alpha\alpha}$	$K_{qq\alpha\beta} = 0$	▷
$= g_{pqqp} : \times -D_{q\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{pp\alpha\alpha}$	$K_{pp\alpha\beta} = 0$	▷
$= g_{qqpp} : \times -D_{p\gamma\gamma'} \mapsto K_{q\gamma\gamma'}$:	$K_{qp\alpha\alpha} = K_{pq\alpha\alpha}^*$	$K_{qp\alpha\beta} = -K_{pq\alpha\beta}$	
$= g_{prpr} : \times -D_{p\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{pr\alpha\alpha}$	$K_{pr\alpha\beta}$	◀
$= g_{rprp} : \times -D_{r\gamma\gamma'} \mapsto K_{r\gamma\gamma'}$:	$K_{rp\alpha\alpha} = K_{pr\alpha\alpha}^*$	$K_{rp\alpha\beta} = -K_{pr\alpha\beta}$	
$= g_{pppp} : \times -D_{p\gamma\gamma'} \mapsto K_{p\gamma\gamma'}$:	$K_{pp\alpha\alpha}$	$K_{pp\alpha\beta} = 0$	▷

Tab. 7.2: Construction of the auxiliary matrices $\mathbf{k}_{\alpha\alpha}, \mathbf{k}_{\alpha\beta} \in \mathbb{C}^{n \times n}$: Only the non-redundant, \blacktriangleleft - and \triangleleft -labeled contributions of tab. 7.1 are considered. Note the factor of $\frac{1}{2}$ multiplied into the density matrix diagonal elements $D_{pp\gamma\gamma'}$ to prevent double counting the contributions to the $\mathbf{K}_{\alpha\alpha}$ diagonal according to eq. 7-2.

	$\mathbf{k}_{\alpha\alpha} \in \mathbb{C}^{n \times n}$	$\mathbf{k}_{\alpha\beta} \in \mathbb{C}^{n \times n}$
$g_{prqs} : \times -D_{qr\gamma\gamma'} \mapsto k_{ps\gamma\gamma'} :$	$k_{ps\alpha\alpha} = k_{sp\alpha\alpha}^*$	$k_{ps\alpha\beta} = -k_{sp\alpha\beta}$
$= g_{qrps} : \times -D_{pr\gamma\gamma'} \mapsto k_{qs\gamma\gamma'} :$	$k_{qs\alpha\alpha} = k_{sq\alpha\alpha}^*$	$k_{qs\alpha\beta} = -k_{sq\alpha\beta}$
$= g_{psqr} : \times -D_{qs\gamma\gamma'} \mapsto k_{pr\gamma\gamma'} :$	$k_{pr\alpha\alpha} = k_{rp\alpha\alpha}^*$	$k_{pr\alpha\beta} = -k_{rp\alpha\beta}$
$= g_{qspr} : \times -D_{ps\gamma\gamma'} \mapsto k_{qr\gamma\gamma'} :$	$k_{qr\alpha\alpha} = k_{rq\alpha\alpha}^*$	$k_{qr\alpha\beta} = -k_{rq\alpha\beta}$
$g_{prqr} : \times -D_{qr\gamma\gamma'} \mapsto k_{pr\gamma\gamma'} :$	$k_{pr\alpha\alpha} = k_{pr\alpha\alpha}^*$	$k_{pr\alpha\beta} = -k_{pr\alpha\beta}$
$= g_{qrpr} : \times -D_{pr\gamma\gamma'} \mapsto k_{qr\gamma\gamma'} :$	$k_{qr\alpha\alpha} = k_{rq\alpha\alpha}^*$	$k_{qr\alpha\beta} = -k_{rq\alpha\beta}$
$g_{prps} : \times -D_{pr\gamma\gamma'} \mapsto k_{ps\gamma\gamma'} :$	$k_{ps\alpha\alpha} = k_{ps\alpha\alpha}^*$	$k_{ps\alpha\beta} = -k_{ps\alpha\beta}$
$= g_{pspr} : \times -D_{ps\gamma\gamma'} \mapsto k_{pr\gamma\gamma'} :$	$k_{pr\alpha\alpha} = k_{rp\alpha\alpha}^*$	$k_{pr\alpha\beta} = -k_{rp\alpha\beta}$
$g_{ppqq} : \times -D_{qp\gamma\gamma'} \mapsto k_{pq\gamma\gamma'} :$	$k_{pq\alpha\alpha} = k_{qp\alpha\alpha}^*$	$k_{pq\alpha\beta} = -k_{qp\alpha\beta}$
$= g_{qppq} : \times -\frac{1}{2}D_{pp\gamma\gamma'} \mapsto k_{qq\gamma\gamma'} :$	$k_{qq\alpha\alpha}$	
$= g_{pqqp} : \times -\frac{1}{2}D_{qq\gamma\gamma'} \mapsto k_{pp\gamma\gamma'} :$	$k_{pp\alpha\alpha}$	
$g_{prpr} : \times -D_{pr\gamma\gamma'} \mapsto k_{pr\gamma\gamma'} :$	$k_{pr\alpha\alpha} = k_{rp\alpha\alpha}^*$	$k_{pr\alpha\beta} = -k_{rp\alpha\beta}$
$g_{pppp} : \times -\frac{1}{2}D_{pp\gamma\gamma'} \mapsto k_{pp\gamma\gamma'} :$	$k_{pp\alpha\alpha}$	

eration of the hermiticity and antisymmetry of the $\mathbf{K}_{\alpha\alpha}$ and $\mathbf{K}_{\alpha\beta}$ blocks, respectively: In tab. 7.1, the unlabeled redundant matrix elements can be generated from the non-redundant, \blacktriangleleft - and \triangleleft -labeled matrix elements from $K_{pq\alpha\alpha} = K_{qp\alpha\alpha}^*$ and $K_{pq\alpha\beta} = -K_{qp\alpha\beta}$. The \triangleleft -labeled entries non-redundant for $\mathbf{K}_{\alpha\alpha}$, but redundant for the $\mathbf{K}_{\alpha\beta}$ block because $K_{pp\alpha\beta} = 0$ for all p .

Tab. 7.2 illustrates a simplified scheme via the auxiliary matrices $\mathbf{k}_{\alpha\alpha}$ and $\mathbf{k}_{\alpha\beta}$, and

$$\mathbf{K}_{\alpha\alpha} =: \mathbf{k}_{\alpha\alpha} + \mathbf{k}_{\alpha\alpha}^{\text{T}*}, \quad \mathbf{K}_{\alpha\beta} =: \mathbf{k}_{\alpha\beta} - \mathbf{k}_{\alpha\beta}^{\text{T}}. \quad (7-2)$$

Employing this simplified scheme is, compared to the straightforward assembly following tab. 7.1, expected to reduce the floating point operation count of `assemble_2cSCF2ParticleParts` by almost a factor of 2. However, because

of the significant, approximately constant overhead from both the construction `CGBTTree_HermitianTupel4_Iterator` and the `switch` block checking for 4-index equalities, `assemble_2cSCF2ParticleParts` CPU times are dominated by floating point operations for large numbers of 4-index integrals only. Thus, significantly accelerated assembly of $\mathbf{G} = \mathbf{J} - \mathbf{K}$ is observed only for large numbers of 4-index integrals, i.e. comparably large basis sets, as showcased for the simple TIH example in fig. 7.1.^b

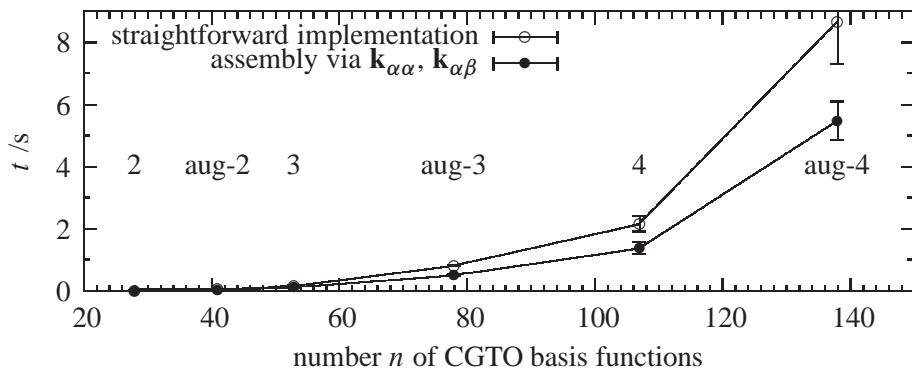


Fig. 7.1: Mean `assemble_2cSCF2ParticleParts` CPU times t : TIH at experimental $d_{\text{Ti-H}} = 3.5338$ a.u., ^{138}Ti small-core MC-DHF PP¹³¹ with cc- and aug-cc-pVXZ basis,^{139,140} for cardinality numbers X of 2, 3, and 4. CPU times t have been averaged over the 23, 22, 25, 24, 27, and 26 SCF iterations run, respectively ($\delta_E = 1 \cdot 10^{-9}$ a.u. energy threshold); error bars are standard deviations.

^b It should be noted that this performance gain is not significant within the conventional HF SCF calculation scheme of the present QOL implementation, which is exclusively dominated by the computation of all (symmetry-unique) 4-index integrals with respect to both CPU time and memory requirements.

Clearly, any attempt to further accelerate assembly of $\mathbf{G} = \mathbf{J} - \mathbf{K}$ requires, in the very first place, moving to an integral-direct^{38,105} or “semi-direct”¹²¹ HF SCF calculation scheme, thus enabling integral screening,¹²¹ \mathbf{D} and \mathbf{G} extrapolation,¹²¹ and RI techniques.^{136,137}

7.2 Eigenvector and Density Matrix Processing

The choice of a 2-spinor basis of N Kramers pairs introduced in sec. 3.1.3, eq. 3-21, i.e.

$$(|\psi_1\rangle, \dots |\psi_{2N}\rangle) = \bigcup_{i=1}^N (|\psi_i\rangle, |\bar{\psi}_i\rangle)_i =: \mathcal{D}$$

does not only place a variational restriction but, as pointed out in sec. 3.1.3, also employs a special unitary symplectic transformation within every eigenspace $\mathbb{E}(\epsilon_i)$. However, whereas there is no physical reason for a particular choice, from an algorithmic point of view it is necessary to define the $2n$ vector representations \mathbf{C}_i such that

$$\mathbf{C}_{i+n} = \bar{\mathbf{C}}_i := (\mathbf{j} \otimes \mathbf{1}_n) \mathbf{C}_i^*, \quad (7-3)$$

i.e. such that eq. 3-31 holds. It is only through eq. 7-3 that any exploration of time reversal-invariance becomes possible.

Stated globally, Roothaan–Hall eigenvector processing – i.e. Fock matrix orthogonalization, diagonalization, back-transformation to \mathbf{C} , construction of a time reversal-invariant basis \mathcal{D} , and assembly of the density matrix $\mathbf{D}(\mathbf{C})$ – has been wrapped in the `HermitianTimeReversalRoothaanHall2cSCF_EigenSystem` class briefly introduced in sec. 7.2.1. The particular transformation techniques applied to arrive at eq. 7-3, and (atomic fractional occupation number) density matrix assembly are explained in sec.s 7.2.2 and 7.2.3, respectively.

7.2.1 Top-Level Self-Consistent Field Algorithm Classes

`HermitianTimeReversalRoothaanHall2cSCF_EigenSystem` is derived from `Hermitian_EigenSystem`, particularly for eigenvector sorting, transformation, and density matrix assembly. It is a single-parameter template class of a Unitary-Space-type argument defining the CGTO-spanned subspace \mathbb{B} the 2-component HF SCF equations are solved in.^c

Generally, the `HermitianTimeReversalRoothaanHall2cSCF_EigenSystem` constructor from two `const HermitianTimeReversalInvariantMat-`

^c An essentially analog `RoothaanHall1cSCF_EigenSystem` class has been implemented to provide a similar interface for the scalar-relativistic spin-free case.

rixRepresentation & and one **unsigned int** & argument, corresponding to **F**, **S**, and the number $2N$ of electrons, respectively, calls the base class constructor and, thus, the LAPACK/BLAS generalized eigenvalue equation solver.^d Central for the communication of eigenvectors \mathbf{C}_i , eigenvalues ϵ_i , and 2-spinor occupation numbers v_i is the std::vector of KramersPairsOf_Molecular2SpinorRepresentations attribute, that is constructed in the course of the transformation of **C** to a time reversal-invariant basis \mathcal{D} .

Both the KramersPairOf_Molecular2SpinorRepresentations and Molecular2SpinorRepresentation are templates of UnitarySpace that allow structured processing of the \mathbf{C}_i , ϵ_i , and v_i : KramersPairOf_Molecular2SpinorRepresentations owns a Molecular2SpinorRepresentation_fwd, and two **double** attributes **_e** and **_n** that mimick the $(\mathbf{C}_i, \epsilon_i, v_i)$ triple; the “time-reverse” 2-spinor vector representation $\bar{\mathbf{C}}_i$ is not held in memory, but generated from **_fwd** if required. The member **SU2Transform()** wraps the transformation of \mathbf{C}_i and $\bar{\mathbf{C}}_i$ to have “maximum similarity” with non-relativistic spin orbitals as discussed in sec. 7.2.2.

Molecular2SpinorRepresentation has been implemented in analogy to the existing matrix representation classes:

```
1 template <class UnitarySpace>
2 class Molecular2SpinorRepresentation : public
3 public QOL::MatrixVector::Matrix<typename
4     UnitarySpace:::_ScalarProduct:::_ScalarProduct_Codomain>
5 {
6 public:
7 // ...
8 const typename
9     UnitarySpace:::_ScalarProduct:::_ScalarProduct_Codomain &
10    operator [] (const int & i) const {
11        return (*this)(i,0);
12    }
13 Molecular2SpinorRepresentation TimeReversalTransform()
14    const;
15 // ...
16 private:
```

^d An optional fourth constructor argument defines a set of possibly fractional atomic occupation numbers different from an Aufbau occupation scheme.

```
17 const UnitarySpace * _space;
18 };
```

Molecular2SpinorRepresentation provides, outside an one-index [] operator for vector entry access and TimeReversalTransform(), special methods for inner and dyadic products, real and imaginary parts, etc.^e

7.2.2 Choosing Time Reversal-Invariant Eigenspace Bases

Operator matrix representations in a basis \mathcal{D} of Kramers pairs, i.e. a basis satisfying eq. 7-3, are necessarily of the special Cayley form of eq. 5-1 from the discussion in sec. 3.2.2. However, the converse is generally not true: If only $\mathbf{F} \in \mathbb{A}_n$ hermitian, the relation

$$\mathbf{FC} = \mathbf{C}\epsilon, \quad (7-4)$$

with $\epsilon = \text{diag } \epsilon_i \in \mathbb{R}^{2n \times 2n}$, does not guarantee \mathbf{C} to be a matrix of n Kramers pairs of eigenvectors $\mathbf{C}_{i+n} = \bar{\mathbf{C}}_i$. Instead, eq. 7-4 defines all $2n$ eigenvectors \mathbf{C}_i up to phase factors ι_i only, which, since generally $\iota_i \notin \mathbb{R}$, do not commute with the (co-representation of the) time reversal operator. Assuming the existence of a matrix $\mathbf{C}' \in \mathbb{A}_n$ of n Kramers pairs $\mathbf{C}'_{i+n} = \bar{\mathbf{C}}'_i$, and writing $\mathbf{C}_i = \iota_i \mathbf{C}'_i$, generally it is $\iota_{i+n} \neq \iota_i^*$

$$\Rightarrow \mathbf{C}_{i+n} \neq \bar{\mathbf{C}}_i. \quad (7-5)$$

One can, however, find the $2n$ inverse phase factors ι_i^{-1} , i.e. re-phase all $2n$ eigenvectors \mathbf{C}_i to $\iota_i^{-1} \mathbf{C}_i = \mathbf{C}'_i$, from

$$\langle \mathbf{C}_{i+n} \iota_{i+n}^{-1}, \iota_i^{*-1} \bar{\mathbf{C}}_i \rangle = \iota_{i+n}^{*-1} \iota_i^{*-1} \langle \mathbf{C}_{i+n}, \bar{\mathbf{C}}_i \rangle \stackrel{!}{=} \langle \mathbf{C}'_{i+n}, \bar{\mathbf{C}}'_i \rangle = 1 \quad (7-6)$$

$$\Rightarrow \iota_{i+n} = \iota_i = \langle \mathbf{C}_{i+n}, \bar{\mathbf{C}}_i \rangle^{-1/2} \quad (7-7)$$

for normalized eigenvectors.

Analog to sec. 3.1.3, a basis of eigenvectors \mathbf{C}_i of a Cayley matrix $\mathbf{F} \in \mathbb{A}_n$ will, in the following, be referred to as “Kramers-restricted”. If, moreover, a set of n phase factor relations is defined such that eq. 7-3 holds for all i , this Kramers-restricted basis of n Kramers pairs $(\mathbf{C}_i, \mathbf{C}_{i+n}) = (\mathbf{C}_i, \bar{\mathbf{C}}_i)$ will be referred to as “Kramers-conjugate” or “time reversal-invariant”.

^e The `l_SubspaceNormSquare(const int &)` and `removeSymmetryContamination(const int &)` methods apply to the special atomic case and are discussed in sec. 7.2.3.

“Kramers Pairing”: Choosing Time Reversal-Invariant Eigenspace Bases

In a straightforward approach, a Kramers-conjugate basis \mathcal{D} could be constructed by iteration over all eigenspaces $\mathbb{b}_i = \text{span}(\mathbf{C}_i, \mathbf{C}_{i+n})$ and re-phasing according to eq. 7-7.

However, for $\dim \mathbb{b}_i = 2n_i > 2$, the LAPACK/BLAS generalized eigenvalue equation solver generally does not provide a basis $D_i = (\mathbf{C}_{i_\nu}, \mathbf{C}_{i_\nu+n})_{\nu=1}^{n_i}$ closed under time reversal: Clearly, for any $\mathbf{C}_{i_\nu} \in D_i$

$$(\mathbf{j} \otimes \mathbf{1}_n) \mathbf{C}_{i_\nu}^* \in \mathbb{b}_i \quad \not\Rightarrow \quad \iota \cdot (\mathbf{j} \otimes \mathbf{1}_n) \mathbf{C}_{i_\nu}^* \in D_i, \quad (7-8)$$

i.e. one cannot always find a phase ι such that the time reversal-transformed basis vector $\bar{\mathbf{C}}_{i_\nu}$ is also a vector of the basis D_i ; but this has been assumed for the re-phasing procedure of eq. 7-7. Although this LAPACK/BLAS behavior has not been observed for the most common $\dim \mathbb{b}_i = 2$ case, higher (for example atomic m_j) degeneracies and, thus, eigenspace basis definitions as those discussed cannot be ruled out *a priori*.

Therefore, a procedure different from the straightforward approach of eq. 7-7 has been adopted:

TimeReversalInvariantRoothaanHall2cSCF_EigenSystem has a **private** method **void construct_KramersConjugate2SpinorBasis()** called by the constructor, i.e. right after the LAPACK/BLAS generalized eigenvalue equation solver has been run via the in-list constructor of the Hermitian_EigenSystem base.

Looping over all eigenspaces \mathbb{b}_i , **construct_KramersConjugate2SpinorBasis()** constructs Kramers-conjugate bases via eq. 7-7 for $\dim \mathbb{b}_i = 2$, and employs a Schmidt-like orthogonalization scheme¹¹¹ for the $\dim \mathbb{b}_i > 2$ case. Along these lines, eigenvectors \mathbf{C}_i and eigenvalues ϵ_i are re-ordered to match eq. 7-3, i.e., for $i \in \{1, \dots, n\}$, from an $\epsilon_{2i-1} = \epsilon_{2i}$ ordering to an $\epsilon_i = \epsilon_{i+n}$ ordering, and the time-forward member of the transformed eigenvector pair $(\mathbf{C}'_i, \bar{\mathbf{C}}'_i)$ is cast to the vector of KramersPairsOf_Molecular2SpinorRepresentations attribute. Thus,

- if $\dim \mathbb{b}_i = 2n_i = 2$, \mathbf{C}_i and \mathbf{C}_{i+n} are re-phased according to eq. 7-3, and that eigenvector with the largest α -spin component is cast to the `_fwd` at-

tribute of the respective KramersPairOf_Molecular2SpinorRepresentations.

- if $\dim \mathbb{b}_i = 2n_i > 2$, an eigenspace basis \mathcal{D}_i of Kramers pairs $(\mathbf{C}'_{i_v}, \bar{\mathbf{C}}'_{i_v})$ is constructed from the basis $D_i = (\mathbf{C}_{i_v}, \mathbf{C}_{i_v+n})_v$ as follows:

In the μ -th step, the projector

$$\begin{aligned} \mathbf{P}^\mu &= \mathbf{P}^{\mu-1} - \frac{1}{\|\mathbf{C}_{i_\mu}\|^2} \mathbf{C}_{i_\mu} \mathbf{C}_{i_\mu}^T - \frac{1}{\|\bar{\mathbf{C}}_{i_\mu}\|^2} \bar{\mathbf{C}}_{i_\mu} \bar{\mathbf{C}}_{i_\mu}^T \\ &= \mathbf{P}^{\mu-1} - \frac{1}{\|\mathbf{C}_{i_\mu}\|^2} \mathbf{C}_{i_\mu} \mathbf{C}_{i_\mu}^T - \frac{1}{\|\mathbf{C}_{i_\mu}\|^2} (\mathbf{j} \otimes \mathbf{1}_{2n}) \mathbf{C}_{i_\mu}^* \mathbf{C}_{i_\mu}^T (\mathbf{j} \otimes \mathbf{1}_{2n})^T \end{aligned} \quad (7-9)$$

is constructed from

$$\mathbf{C}_{i_\mu} := \arg \max_v \left\{ \|\mathbf{C}_{i_v}\|^2 \mid \mathbf{C}_{i_v} \in D_i^{\mu-1} \right\}, \quad (7-10)$$

$$D_i^{\mu-1} := (\mathbf{P}^{\mu-1} \dots \mathbf{P}^1 \mathbf{C}_{i_v}, \mathbf{P}^{\mu-1} \dots \mathbf{P}^1 \mathbf{C}_{i_v+n})_v \setminus \{\mathbf{0}\} \quad (7-11)$$

and, subsequently, applied to all vectors in $D_i^{\mu-1}$ to give a new $D_i^\mu = (\mathbf{P}^\mu D_i^{\mu-1}) \setminus \{\mathbf{0}\}$. The vectors $\mathbf{C}_{i_\mu}, \bar{\mathbf{C}}_{i_\mu}$ are re-normalized to $\mathbf{C}'_{i_\mu}, \bar{\mathbf{C}}'_{i_\mu}$ and added to

$$\mathcal{D}_i^{\mu-1} := \bigcup_{v=1}^{\mu-1} (\mathbf{C}'_{i_v}, \bar{\mathbf{C}}'_{i_v}) \quad (7-12)$$

to give a new \mathcal{D}_i^μ . Technically, in every step μ , that member of the pair added to \mathcal{D}^μ with the largest α -spin component norm is cast to the `_fwd` attribute of KramersPairOf_Molecular2SpinorRepresentations.

After n_i steps, $D_i^{n_i} = \emptyset$ and $\mathcal{D}_i^{n_i} =: \mathcal{D}_i$ is the sought eigenspace basis of n_i Kramers pairs $(\mathbf{C}'_{i_v}, \bar{\mathbf{C}}'_{i_v})$.

Note that, in the $2n_i > 2$ case, no re-phasing step is necessary for eq. 7-3 to hold, as this relation is built in the Schmidt-like procedure: In the μ -th step, \mathcal{D}^μ is constructed with reference to only the largest-norm vector $\mathbf{C}_{i_\mu} \in D_i^{\mu-1}$, whereas the second member of \mathcal{D}^μ is *defined* as $\bar{\mathbf{C}}_{i_\mu}$.

The remaining degree of freedom, i.e. the special unitary symplectic transformation among the members of every Kramers pair $(\mathbf{C}'_i, \bar{\mathbf{C}}'_i)$, is removed by choosing a set of particular transformation to maximize the 2-spinors' similarity to non-relativistic spin orbitals.²⁸

Choosing Maximum 2-Spinor Similarity to Non-Relativistic Spin Orbitals

As briefly discussed in sec. 3.1.3, a given Kramers pair $(\mathbf{C}'_i, \bar{\mathbf{C}}'_i)$ of vector representations of 2-spinors is defined up to a special unitary symplectic transformation only. Precisely, one is allowed to choose $v_i, w_i, \iota_i \in \mathbb{C}$ such that

$$(\mathbf{C}'_i, \bar{\mathbf{C}}'_i)_i \mapsto (\mathbf{C}'''_i, \bar{\mathbf{C}}'''_i)_i := (\iota_i(v_i \mathbf{C}'_i + w_i \bar{\mathbf{C}}'_i), \iota_i^*(-w_i^* \mathbf{C}'_i + v_i^* \bar{\mathbf{C}}'_i))_i, \quad (7-13)$$

$$|v_i|^2 + |w_i|^2 = 1, \quad |\iota_i|^2 = 1, \quad (7-14)$$

symbolically

$$(\mathbf{C}'''_i, \bar{\mathbf{C}}'''_i) = (\mathbf{C}'_i, \bar{\mathbf{C}}'_i) \begin{pmatrix} \iota_i v_i & -\iota_i^* w_i^* \\ \iota_i w_i & \iota_i^* v_i^* \end{pmatrix}; \quad (7-15)$$

eq. 7-3 also holds for the transformed Kramers pair from the Cayley form of the transposed transformation matrix in eq. 7-15.

Following Hafner and Schwarz,²⁸ the parameters v_i, w_i , and ι_i are chosen as to maximize the 2-spinors' "similarity" to the respective spin orbitals from the non-relativistic, i.e. spin-symmetric limit, by maximizing the norms of the time-forward 2-spinor's α -spin and real part (and of the time-reverse 2-spinor's β -spin component and real part).

Maximizing the norm of the α -spin component of $\mathbf{C}''_i = v_i \mathbf{C}'_i + w_i \bar{\mathbf{C}}'_i$ defines

$$(v_i, w_i) := \arg \max_{v_i, w_i} \{ \| \mathbf{C}_{i\alpha}'' \|^2 - l_i (|v_i|^2 + |w_i|^2 - 1) \} \quad (7-16)$$

from the constraint 7-14, i.e. by making the Lagrangian

$$L_i = \|v_i \mathbf{C}_{i\alpha}'\|^2 - 2\text{Re}(v_i^* w_i \langle \mathbf{C}_{i\alpha}', \mathbf{C}_{i\beta}' \rangle) + \|w_i \mathbf{C}_{i\beta}'\|^2 - l_i (|v_i|^2 + |w_i|^2 - 1)$$

stationary with respect to $u^* \in \{v_i^*, w_i^*\}$; thus,^f

$$\frac{\partial L_i}{\partial u^*} = \begin{cases} v_i \| \mathbf{C}_{i\alpha} \|^2 - w_i \langle \mathbf{C}_{i\alpha}, \mathbf{C}_{i\beta}^* \rangle - l_i v_i \stackrel{!}{=} 0 & \text{for } u^* = v_i^* \\ w_i \| \mathbf{C}_{i\beta}^* \|^2 - v_i \langle \mathbf{C}_{i\beta}^*, \mathbf{C}_{i\alpha} \rangle - l_i w_i \stackrel{!}{=} 0 & \text{for } u^* = w_i^* \end{cases} \quad (7-17)$$

In a similar way, the phase ι_i can be chosen as to maximize the norm of the real part of $\mathbf{C}'''_i = \iota_i \mathbf{C}''_i$, i.e.

^f Differentiation of L_i with respect to v_i and w_i gives the complex-conjugate eq.s 7-17.

$$\iota_i = a_i + ib_i := \arg \max_{a_i, b_i} \left\{ \|\operatorname{Re} \iota_i \mathbf{C}_i''\|^2 - \lambda_i (|\iota_i|^2 - 1) \right\}. \quad (7-18)$$

Expanding $\operatorname{Re} \mathbf{C}_i'' := \frac{1}{2}(\iota_i \mathbf{C}_i'' + \iota_i^* \mathbf{C}_i''')$ gives the Lagrangian

$$\Lambda_i = \|\operatorname{Re} (a_i \operatorname{Re} \mathbf{C}_i' + ia_i \operatorname{Im} \mathbf{C}_i' + ib_i \operatorname{Re} \mathbf{C}_i' - b_i \operatorname{Im} \mathbf{C}_i')\|^2 - \lambda_i (a_i^2 + b_i^2 - 1)$$

which is stationary in terms of the real and imaginary parts a_i and b_i of ι_i if

$$\begin{aligned} & \frac{1}{2} \frac{\partial \Lambda_i}{\partial c} \\ &= \begin{cases} a_i \|\operatorname{Re} \mathbf{C}_i''\|^2 - b_i \langle \operatorname{Re} \mathbf{C}_i'', \operatorname{Im} \mathbf{C}_i'' \rangle - \lambda_i a_i \stackrel{!}{=} 0 & \text{for } c = a_i \\ b_i \|\operatorname{Im} \mathbf{C}_i''\|^2 - a_i \langle \operatorname{Im} \mathbf{C}_i'', \operatorname{Re} \mathbf{C}_i'' \rangle - \lambda_i b_i \stackrel{!}{=} 0 & \text{for } c = b_i \end{cases} \quad (7-19) \end{aligned}$$

using $\operatorname{Re} \mathbf{C}_i'', \operatorname{Im} \mathbf{C}_i'' \in \mathbb{R}^{2n} \Rightarrow \langle \operatorname{Re} \mathbf{C}_i'', \operatorname{Im} \mathbf{C}_i'' \rangle \in \mathbb{R}$. From $\iota_i = e^{i\varphi_i}$, eq.s 7-19 can be re-cast in terms of a single real parameter φ_i such that $a_i = \cos \varphi_i$, $b_i = \sin \varphi_i$.

Therefore, both parameter sets (v_i, w_i) and $(\cos \varphi_i, \sin \varphi_i)$ are obtained from the analog linear systems of equations 7-17 and 7-19.

Technically, the transformation to maximum similarity to non-relativistic spin orbitals was implemented as **void SU2Transform()** method of **KramersPairOf_Molecular2SpinorRepresentations** called for all **KramersPairOf_Molecular2SpinorRepresentations** constructed within **construct_KramersConjugate2SpinorBasis()**. The transformation parameters v_i , w_i , and ι_i are computed analytically from eq.s 7-17 and 7-19.^g

^g The normalized solutions \mathbf{e}^\pm of the hermitian 2×2 eigenvalue problem $\mathbf{t}\mathbf{e}^\pm = \lambda^\pm \mathbf{e}^\pm$ are computed as

$$e_1^\pm = \frac{1}{\sqrt{1 + (\lambda^\pm - t_{11})^2 t_{12}^{-2}}}, \quad e_2^\pm = \frac{\lambda^\pm - t_{11}}{\sqrt{1 + (\lambda^\pm - t_{11})^2 t_{12}^{-2}}}.$$

with $\lambda^\pm = \frac{1}{2}(t_{11} + t_{22}) \pm \sqrt{\frac{1}{4}(t_{11} - t_{22})^2 + |t_{12}|^2} \in \mathbb{R}$ from the characteristic polynomial. Both \mathbf{e}^+ and \mathbf{e}^- are evaluated to assert that the solutions of eq.s 7-17 and 7-19 are not only stationary points, but really maximizers of $\|\mathbf{C}_{i\alpha}''\|^2$ and $\|\operatorname{Re} \mathbf{C}_i'''\|^2$, i.e. satisfy eq.s 7-16 and 7-18, respectively.

For the ease of notation, the three-fold primed time reversal-invariant eigenspace basis vectors similar to non-relativistic spin orbitals \mathbf{C}_i''' will, from here on, be simply written as \mathbf{C}_i , the discussed transformations being understood.

7.2.3 Atomic Occupation Numbers and Angular Symmetry

Most atoms' ground state electronic structure cannot be modeled by a single HF SCF Slater determinant because of its open-shell nature. Whereas unrestricted¹⁴¹ and restricted open-shell¹⁰ formalisms are, to some extent, applicable in a number of special (i.e. J -doublet) cases, this is certainly not true for the Kramers-restricted closed shell HF SCF ansatz of eq. 3-2.

However, from, e.g., the point of view of approximate initial guess molecular from atomic density matrices discussed in sec. 7.4, also very simple approximate models of atomic electronic densities are of some value.

SCF Density Matrices for Fractional Occupations of Atomic 2-Spinors

A straightforward approach is a formal generalization of the SCF density matrix expression of eq.s 3-39 and 3-40 to

$$\mathbf{D} = \mathbf{C} \mathbf{v} \mathbf{C}^{T*} \quad (7-20)$$

$$\Rightarrow \begin{cases} D_{pq\alpha\alpha} = \frac{1}{2} \sum_{i=1}^n (C_{pi\alpha} C_{qi\alpha}^* + C_{pi\beta}^* C_{qi\beta}) \cdot v_i \\ D_{pq\alpha\beta} = \frac{1}{2} \sum_{i=1}^n (C_{pi\alpha} C_{qi\beta}^* - C_{pi\beta}^* C_{qi\alpha}) \cdot v_i \end{cases} \quad (7-21)$$

with the “occupation number matrix”

$$\mathbf{v} = \frac{1}{2} \operatorname{diag} v_i \oplus \frac{1}{2} \operatorname{diag} v_i \in \mathbb{R}^{2n \times 2n} \quad (7-22)$$

of generally non-integer “occupation numbers” $v_i \in [0, 2]$ of the i -th Kramers pair of 2-spinors,^h and $\operatorname{Tr} \mathbf{v} = \sum_{i=1}^n v_i = 2N$. Clearly, eq.s 7-21 collapse to the

^h A less clumsy definition of \mathbf{v} could be written as, e.g., $\mathbf{v} = \operatorname{diag} v_i$, with $v_i \in [0, 1]$ the occupation number of the i -th 2-spinor. However, in the Kramers-restricted closed-shell formalism $v_i = v_{i+n}$ as indicated by eq. 7-22, and v_i can be referred to as a property of the Kramers pair $\mathbf{C}_i, \bar{\mathbf{C}}_i$.

special eq.s 3-39 and 3-40 for a closed-shell occupation number matrix

$$\mathbf{1}_N \oplus \mathbf{0}_{n-N} \oplus \mathbf{1}_N \oplus \mathbf{0}_{n-N}. \quad (7-23)$$

Then, starting with eq. 7-21, the v_i can be set to that fraction of N_{nl} electrons in a given, not necessarily closed, atomic (n, l) shell that “occupies” every of the $2l + 1$ Kramers pairs in that shell, i.e.

$$v_{i_{nl}} = v_{nl} = \frac{N_{nl}}{2l + 1} \quad (7-24)$$

for all Kramers pairs i_{nl} in that shell.

This *ad hoc* choice of atomic density matrices is similar to the Grand Canonical Hartree–Fock and Kohn–Sham (GCHF, GCKS) SCF theories of Abdulnur et al.¹⁴² and Jørgensen and Öhrn,¹⁴³ but the particular choice of v by eq. 7-24 does generally not correspond to physically meaningful ensembles.

For a non-Aufbau occupation,ⁱ the TimeReversalInvariantRoothaanHall2-cSCF_EigenSystem constructor takes an optional fourth **const** string & argument defining this occupation scheme in conventional notation, e.g. "5s^2 5p^6 5d^2 6s^2" for the ground state ^{72}Hf atom with 60-electron PP. Then, a special occupation number-to-eigenvector assignment block is executed:

- The first step involves mapping the blind string data to an internally structured occupation scheme object
- ```
std::map<std::pair<QOL::ShellIndex::Shell, unsigned int>,
 unsigned int> Aufbau;
```

In Aufbau, an atomic  $(n, l)$  shell is modeled by an STL pair of the QOL Shell class and an **unsigned int** main quantum number  $n$ . Shell is the first pair template argument to have atomic shells ordered by increasing orbital angular momentum quantum numbers  $l$  and, within, main quantum numbers  $n$ . The STL map is, then, used to assign an **unsigned int** number  $N_{nl}$  of electrons to all members of the ordered set of atomic shells.

---

<sup>i</sup> The term “Aufbau” or “Aufbau occupation” refers to an atomic occupation scheme obeying the “Aufbau principle”,<sup>144</sup> i.e., for an atom of  $Z$  electrons, to a special occupation scheme that defines the  $Z$  lowest-energy 2-spinors as occupied.<sup>37</sup> In the present implementation, however, the term “Aufbau” is employed somewhat loosely to label any, including non-Aufbau occupation number-to-eigenvector assignments.

- To come to an assignment of these occupation numbers  $\nu_{nl}$  to eigenvectors  $\mathbf{C}_i$ , the latters' orbital angular momentum quantum numbers  $l_i$  are identified by

$$l_i := \arg \max_l \left\{ \sum_{p \in B_l} (|C_{pi\alpha}|^2 + |C_{pi\beta}|^2) \right\}_{li}, \quad (7-25)$$

i.e. by projecting all  $\mathbf{C}_i$  to all subspaces spanned by the CGTO basis functions  $|\phi_p\rangle \in B_l \subseteq B$  with orbital angular momentum quantum numbers  $\|\mathbf{m}_p\|_1 = l$ , and assigning that  $l_i$  that gives rise to the projection with the largest norm-square.

- Finally, iterating over all Aufbau entries, the lowest-energy  $\mathbf{C}_i$  with  $l_i$  equal to the current Aufbau entry's  $l$  is interpreted as belonging to this atomic shell, and the `_n` attribute of the corresponding `KramersPairOf_Molecular2SpinorRepresentations` is set to

$$\nu_i \leftrightarrow \nu_{i_nl} \leftrightarrow \nu_{nl} = \frac{N_{nl}}{2l+1}, \quad (7-26)$$

which effectively enables the connection of the flat 2-spinor index  $i$  and the atomic shell quantum numbers  $n$  and  $l$  implied through eq. 7-24. The same occupation number is assigned to the next  $(2l+1)-1$  eigenvectors  $\mathbf{C}_i$  with the same orbital angular momentum quantum number  $l_i$ ,<sup>j</sup> and the current pair`<Shell, unsigned int>` is removed from Aufbau.

The default reference to an empty string defines an Aufbau occupation by eq. 7-23.

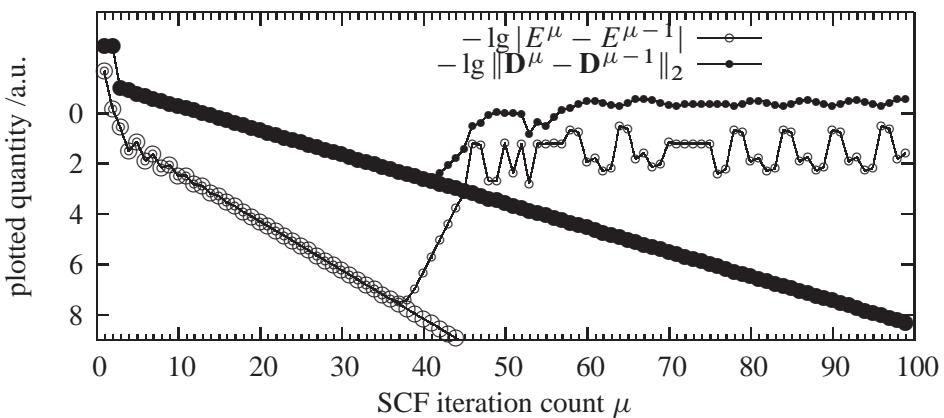
---

<sup>j</sup> Note that the present implementation assumes that  $\epsilon_{nlj} < \epsilon_{n'l'j}$  for  $n < n'$  and all  $j$ , i.e. that, for a given common  $l$ , the  $(n, l + \frac{1}{2})$  2-spinors' energies are still lower than the  $(n', l - \frac{1}{2})$  2-spinors' energies. This might not be true for cases with extreme spin-orbit splitting!<sup>145</sup>

It is noted in passing that it is generally not sufficient to exclude such extreme scenarios from only physical arguments. In early, poorly damped SCF iterations starting from bad initial densities, the “Knotenregel” has been numerically observed to be broken by strong spin-orbit coupling, such that  $\epsilon_{nlj} < \epsilon_{n'l'j}$  for  $n > n'$  and  $l < l'$ : For example, in the the  $^{72}\text{Hf}$  test case illustrated in fig. 7.2, the 4-th SCF iteration  $6p_{1/2}$  virtual 2-spinor energies drop between the  $5d_{3/2}$  and  $5d_{5/2}$  (fractionally) occupied 2-spinor energies. “Knotenregel” violations – within the stated assumption  $\epsilon_{nlj} < \epsilon_{n'l'j}$  for  $n < n'$  and all  $j$  – are, however, routinely handled with the present implementation.

### Atomic Angular Symmetry

Having established this occupation number-to-eigenvector assignment, the assembly of  $\mathbf{D}$  according to eq. 7-21 is straightforward. However, it has been observed in a large number of atomic test cases that convergence of the atomic self-consistent field is not. Instead, convergence is spoiled by contamination of  $\mathbf{C}$  and, thus,  $\mathbf{D}$ .

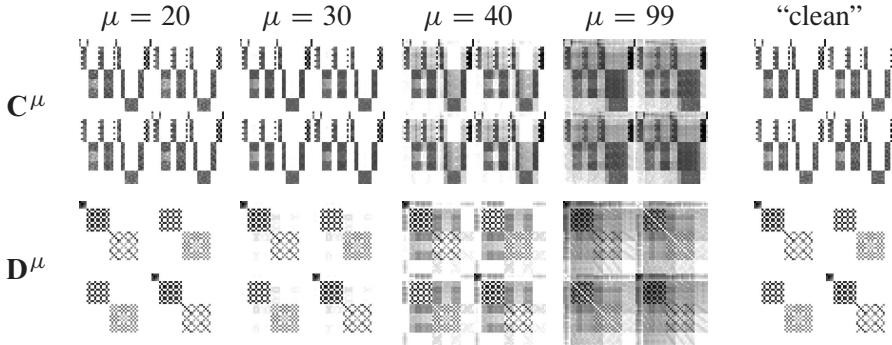


**Fig. 7.2:** Angular symmetry breakdown and `removeSymmetryContamination`: Convergence behavior of HF SCF energy  $E^\mu$  and density matrix  $\mathbf{D}^\mu$  for the  $^{72}\text{Hf}$  atom, small-core MC-DHF PP with cc-pVDZ basis,  $^{146}\text{5s}^2\text{5p}^6\text{5d}^2\text{6s}^2$  occupation scheme, without (small circles  $\circ$ ,  $\bullet$ ) and with `removeSymmetryContamination` (big circles  $\bigcirc$ ,  $\bullet$ )

matrix elements with almost-zero contributions from the numerical solution of the underlying generalized eigenvalue equation systems.

As showcased for the  $^{72}\text{Hf}$  example in fig.s 7.2 and 7.3, this is amplified over the SCF loop: At  $\mu = 37$ , both  $|E^\mu - E^{\mu-1}|$  and  $\|\mathbf{D}^\mu - \mathbf{D}^{\mu-1}\|_2$  begin to increase again with increasing  $\mu$  as, illustrated in fig. 7.3, angular symmetry “blocking” of  $\mathbf{C}$  and  $\mathbf{D}$  breaks down.

Therefore, in every SCF iteration and for all eigenvectors  $\mathbf{C}_i$ , all eigenvector components  $C_{piy}$  with respect to CGTOs  $|\phi_p\rangle$  with  $l_p \neq l_i$  are discarded, and the “clean”, angular symmetry-enforced eigenvectors are re-normalized. Technically, a function `void removeSymmetryContamination(unsigned int &)` was implemented as a `Molecular2SpinorRepresentation` method:



**Fig. 7.3:** Pictorial representations of eigenvector and density matrices  $\mathbf{C}^\mu$  and  $\mathbf{D}^\mu$  from the 20-th, 30-th, 40-th, and 99-th SCF iteration  $\mu$  of fig. 7.2; the rightmost matrix pictures correspond to the “clean”, angular symmetry-enforced 99-th SCF iteration matrices employing `removeSymmetryContamination`.

```

1 template <class USpace>
2 inline void Molecular2SpinorRepresentation<USpace>::
3 removeSymmetryContamination(const int & l)
4 {
5 typename Molecular2SpinorRepresentation<USpace>::_Field
6 _T;
7 typename _T::value_type norm = sqrt(this->norm());
8
9 const int n = _space->basis().size();
10 unsigned int p = 0;
11
12 // ITERATE OVER CGTO BASIS FUNCTIONS:
13 for(CGBTTree_Iterator<typename USpace::_CGBasisType>
14 I(_space->basis()); I.valid(); ++I, ++p)
15 {
16 // first.first.l() RETURNS ANGULAR MOMENTUM QUANTUM NUMBER
17 if(I.i2()->first.first.l() != 1) {
18 (*this)[p] = 0;
19 (*this)[p+n] = 0; }
20 else {
21 (*this)[p] *= norm/sqrt(this->norm());
22 (*this)[p+n] *= norm/sqrt(this->norm()); }
23 }
```

24 }

As soon as a given eigenvector's orbital angular momentum quantum number  $l_i$  is evaluated in the course of the occupation number-to-eigenvector assignment, this eigenvector is cast to `Molecular2SpinorRepresentation`, and its `removeSymmetryContamination` method is called with  $l_i$  as argument.

As evident from the particular example shown in fig.s 7.2 and 7.3, enforcing angular symmetry greatly facilitates, if not enables SCF convergence in the general atomic case. With the exception of oscillations for small  $\mu$ , which are not atypical for calculations starting from poor (e.g., core Hamiltonian) initial guess density matrices, convergence is generally smooth, yet slow.

### 7.3 Optimal Damping

In order to improve SCF convergence also for the general molecular, large basis set case, the 2006 QOL implementation's naïve Roothaan–Hall-like SCF algorithm of sec. 4.3.3 has been abandoned in favor of the Optimal Damping Algorithm (ODA) of Cancés and Le Bris.<sup>36,37</sup>

Stated briefly, the ODA is the most simple of a class of more general “relaxed constraint algorithms” that relax the nonlinear idempotency constraint  $\mathbf{D}^\mu \mathbf{S} \mathbf{D}^\mu = \mathbf{D}^\mu$  over the minimization of  $E_{\text{SCF}}(\mathbf{D})$ ; idempotency is recovered at convergence of  $\mathbf{D}$ . Operationally, in the  $\mu + 1$ -th SCF iteration one does not diagonalize the Fock matrix  $\mathbf{F}^{\mu+1}$ , but, instead,  $\tilde{\mathbf{F}}^{\mu+1} = \mathbf{F}(\tilde{\mathbf{D}}^{\mu+1})$  constructed from the “optimally damped” density matrix

$$\tilde{\mathbf{D}}^{\mu+1} := (1 - \lambda)\tilde{\mathbf{D}}^\mu + \lambda\mathbf{D}^{\mu+1} \quad (7-27)$$

$$\Rightarrow \tilde{\mathbf{F}}^{\mu+1} = (1 - \lambda)\tilde{\mathbf{F}}^\mu + \lambda\mathbf{F}^{\mu+1}. \quad (7-28)$$

From the similarity of eq.s 7-27 and 7-28 to analog expressions defining  $\mathbf{D}$  and  $\mathbf{F}$  matrix damping techniques,<sup>147–149</sup> the parameter

$$\lambda := \arg \min_{\lambda \in [0, 1]} \{E_{\text{SCF}}(\tilde{\mathbf{D}}^{\mu+1})\}, \quad (7-29)$$

is referred to as “optimal damping parameter”.

Both the derivation and the detailed discussion of the SCF ODA, as given by Cancés and Le Bris,<sup>36,37</sup> will not be repeated here.<sup>k</sup> However, the 2-component HF SCF energy functional of time reversal-invariant,  $\mathbb{C}^{2n \times 2n}$  density matrices  $\mathbf{D}$ , i.e.

$$E_{\text{SCF}}(\mathbf{D}) = \text{Tr} (\mathbf{h} + \frac{1}{2}\mathbf{G}(\mathbf{D}))\mathbf{D} \quad (7-30)$$

differs from the 1-component HF SCF energy functional of spin-restricted,  $\mathbb{R}^{n \times n}$  density matrices  $\mathbf{D}$  considered by Cancés and Le Bris. Consequently, the analytic expression for the optimal damping parameter  $\lambda$  obtained via eq. 7-29 is slightly different:

Abbreviating  $\tilde{E}_{\text{SCF}}^{\mu+1} = E_{\text{SCF}}(\tilde{\mathbf{D}}^{\mu+1})$ , from eq.s 7-30 and 7-27,

$$\begin{aligned} & \tilde{E}_{\text{SCF}}^{\mu+1} \\ &= \tilde{E}_{\text{SCF}}^\mu + \lambda \text{Tr} (\tilde{\mathbf{F}}^\mu (\mathbf{D}^{\mu+1} - \tilde{\mathbf{D}}^\mu)) + \frac{1}{2}\lambda^2 \text{Tr} ((\mathbf{F}^{\mu+1} - \tilde{\mathbf{F}}^\mu)(\mathbf{D}^{\mu+1} - \tilde{\mathbf{D}}^\mu)) \\ &= \tilde{E}_{\text{SCF}}^\mu + \lambda a^{\mu+1} + \frac{1}{2}\lambda^2 b^{\mu+1} \end{aligned}$$

such that eq. 7-29 implies

$$\lambda^{\mu+1} = \begin{cases} 1 & \text{if } a^{\mu+1} \leq -b^{\mu+1} \\ -b^{\mu+1}/a^{\mu+1} & \text{else} \end{cases} \quad (7-31)$$

which differs from the 1-component Cancés–Le Bris expression in a factor of  $\frac{1}{2}$  for  $a^{\mu+1}$ .

It has been observed that, in late SCF iterations employing tight convergence thresholds, computation of the optimal damping parameter  $\lambda^{\mu+1}$  becomes numerically unstable as both  $\mathbf{D}^{\mu+1} - \tilde{\mathbf{D}}^\mu \rightarrow \mathbf{0}_{2n}$  and  $\mathbf{F}^{\mu+1} - \tilde{\mathbf{F}}^\mu \rightarrow \mathbf{0}_{2n}$  with increasing  $\mu$ . Therefore, in the present implementation  $a^{\mu+1}$  and  $b^{\mu+1}$  are computed from

<sup>k</sup> Put in a somewhat simplified way, the ODA is a minimization scheme of  $E_{\text{SCF}}(\mathbf{D})$  on the set  $\tilde{\Delta} = \{\tilde{\mathbf{D}} \in \mathbb{C}^{2n \times 2n} \mid \text{Tr} \tilde{\mathbf{D}} = 2N, \|\tilde{\mathbf{D}}\| \leq \|\tilde{\mathbf{D}}\|\}$  of “relaxed constraint” density matrices. Clearly,  $\tilde{\Delta}$  contains the set  $\Delta$  of all idempotent density matrices  $\mathbf{D}$  as proper subset.

The ODA is motivated and enabled by the facts that, first,  $E_{\text{SCF}}$  has the same minima on  $\tilde{\Delta}$  and  $\Delta \subset \tilde{\Delta}$ ,<sup>37</sup> and, second, that  $\tilde{\Delta}$  is convex, i.e. from  $\{\tilde{\mathbf{D}}_i\}_i \subseteq \tilde{\Delta} \Rightarrow \sum_i \lambda_i \tilde{\mathbf{D}}_i \in \tilde{\Delta}$  if only  $\sum_i \lambda_i = 1$  for non-negative  $\lambda_i$ .<sup>37</sup>

Then, in each SCF iteration  $\mu$ , diagonalization of  $\mathbf{F}(\tilde{\mathbf{D}}^\mu)$  gives  $\mathbf{D}^{\mu+1} \in \Delta$  – defining the “steepest descent” of  $E_{\text{SCF}}(\tilde{\mathbf{D}}^\mu)$  in the direction of idempotent density matrices  $\mathbf{D} \in \Delta$  – that serves to construct a new input  $\tilde{\mathbf{D}}^{\mu+1} \in \tilde{\Delta}$  via eq. 7-27, with  $\lambda$  defined such that  $E_{\text{SCF}}$  is minimized.

the 1- and 2-electron energies  $E_1^{\mu+1}$  and  $E_2^{\mu+1}$ , the 2-spinor energy matrix  $\epsilon^{\mu+1}$ , and 2-spinor occupation number matrix  $\mathbf{v}^{\mu+1}$  via

$$a^{\mu+1} = E_1^{\mu+1} + 2E_2^{\mu+1} - b^\mu - \text{Tr } \mathbf{F}^{\mu+1} \tilde{\mathbf{D}}^\mu, \quad (7-32)$$

$$b^{\mu+1} = \text{Tr } \epsilon^{\mu+1} \mathbf{v}^{\mu+1} - \text{Tr } \tilde{\mathbf{F}}^\mu \tilde{\mathbf{D}}^\mu \quad (7-33)$$

with

$$E_1^{\mu+1} = \text{Tr } \mathbf{h} \mathbf{D}^{\mu+1}, \quad 2E_2^{\mu+1} = \text{Tr } \mathbf{G}^{\mu+1} \mathbf{D}^{\mu+1} \quad (7-34)$$

to avoid repeated computation of traces (of products) of almost-zero matrices.<sup>1</sup>

The implemented 2-component SCF ODA can thus be stated as follows:

- construction of  $\mathbf{S}$ ,  $\mathbf{T}$ ,  $\mathbf{U}$ , and  $\mathbf{W}$  matrices;
- evaluation of all symmetry-unique electron–electron repulsion 4-index integrals  $g_{prqs} \in \mathcal{G}_n$ , and storage;
- setting the initial guess density matrix  $\mathbf{D}^0$  to null;  
then, for  $\delta_{\mathbf{D}} > 0$  a density threshold, in the  $\mu$ -th SCF iteration

**1:** orthogonalize  $\tilde{\mathbf{F}}^\mu$ , diagonalize, and back-transform to obtain the  $\mu$ -th eigenvector matrix  $\mathbf{C}^\mu$ ;

construct a Kramers-conjugate basis and assemble a new density matrix  $\mathbf{D}^{\mu+1}$  via eq. 7-21;

**2:** if  $\|\mathbf{D}^{\mu+1} - \mathbf{D}^\mu\|_2 \leq \delta_{\mathbf{D}}$  terminate and compute  $E_{\text{SCF}}^{\mu+1}$ ;  
else

**3:** assemble  $\mathbf{G}^{\mu+1} = \mathbf{G}(\mathbf{D}^{\mu+1})$  via eq.s 3-41 and 3-42;

**4:** assemble the  $\mu + 1$ -th Fock matrix  $\mathbf{F}^{\mu+1} = \mathbf{F}(\mathbf{D}^{\mu+1}) = \mathbf{h} + \mathbf{G}^{\mu+1}$ ;

**5:** compute  $E_1^{\mu+1}$ ,  $E_2^{\mu+1}$ ,  $a^{\mu+1}$ ,  $b^{\mu+1}$  according to eq.s 7-34, 7-32, and 7-33, respectively, and the  $\mu + 1$ -th optimal damping parameter  $\lambda^{\mu+1}$  as

$$\lambda^{\mu+1} = \begin{cases} 1 & \text{if } a^{\mu+1} \leq -b^{\mu+1} \\ -b^{\mu+1}/a^{\mu+1} & \text{else} \end{cases}$$

---

<sup>1</sup> Moreover, if  $a^{\mu+1}$  and/or  $b^{\mu+1}$  drop below a threshold  $\delta$ , typically set to  $\delta = 10^{-12}$ ,  $\lambda^{\mu+1}$  is set to  $\frac{1}{2}$ , defining a simple averaging of  $\tilde{\mathbf{D}}^\mu$  and  $\mathbf{D}^{\mu+1}$ .

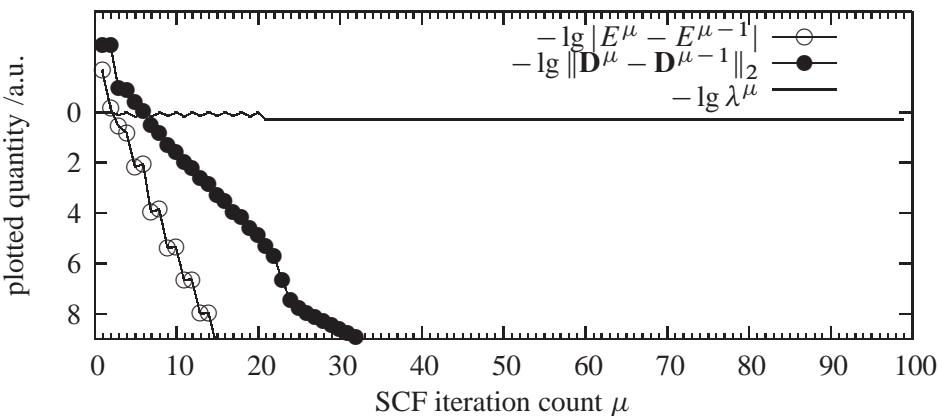
**6:** assemble

$$\tilde{\mathbf{D}}^{\mu+1} := (1 - \lambda^{\mu+1})\tilde{\mathbf{D}}^\mu + \lambda^{\mu+1}\mathbf{D}^{\mu+1}$$

$$\tilde{\mathbf{F}}^{\mu+1} := (1 - \lambda^{\mu+1})\tilde{\mathbf{F}}^\mu + \lambda^{\mu+1}\mathbf{F}^{\mu+1}$$

and go to 1.

Fig. 7.4 shows, for the  $^{72}\text{Hf}$  atom example, the improved SCF convergence behavior, as compared to the performance of the Roothaan–Hall-type SCF algorithm employed for fig. 7.2.  $E_{\text{SCF}}$  is converged below  $10^{-9}$  a.u. in only 15, as

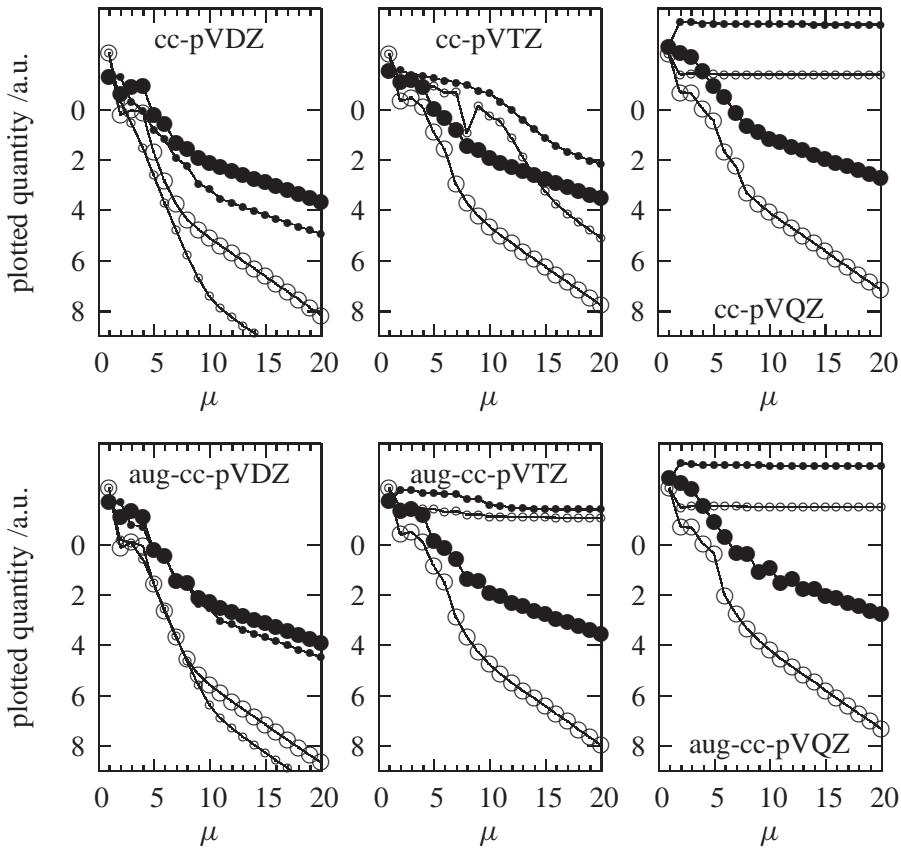


**Fig. 7.4:** Performance of the SCF ODA for the  $^{72}\text{Hf}$  atom, small-core MC-DHF PP with cc-pVQZ basis,  $^{146}\text{S}^2 5\text{p}^6 5\text{d}^2 6\text{s}^2$  occupation scheme. The same plot scale as in fig. 7.2 has been chosen for the ease of comparison with the Roothaan–Hall-type SCF algorithm.

opposed to 44 iterations.

Although tempting to conclude from comparison of fig.s 7.2 and 7.4 only, the SCF ODA does generally not *accelerate* convergence. In fact, algorithms employing convergence acceleration techniques as, e.g., DIIS,<sup>150</sup> and advanced relaxed constraint algorithms<sup>36,151,152</sup> have been shown to be superior to the SCF ODA in late iterations near stationary points of  $E_{\text{SCF}}$ .<sup>36</sup> However, the SCF ODA has been shown to be more stable, particularly in early iterations,<sup>36</sup> and is guaranteed to converge to a minimum of the HF SCF energy.<sup>37m</sup>

<sup>m</sup>Whether this remarkable, yet formal algorithmic property is of practical value from the point of



**Fig. 7.5:** Comparison of Roothaan–Hall-type and SCF ODA convergence behavior: TIH at experimental  $d_{\text{Ti}-\text{H}} = 3.5338 \text{ a.u.}$ ,<sup>138</sup>  $^{81}\text{Ti}$  small-core MC-DHF PP<sup>131</sup> with cc-pV $X$ Z and aug-cc-pV $X$ Z basis,<sup>139,140</sup> for cardinality numbers  $X$  of 2, 3, and 4. Displayed are both  $-\lg |E^\mu - E^{\mu-1}|$  and  $-\lg \|\mathbf{D}^\mu - \mathbf{D}^{\mu-1}\|_2$  for the Roothaan–Hall-type (small circles ○, ●) and SCF ODA (big circles ○, ●), respectively.

These properties are, to some extent, illustrated by the seemingly uncomplicated TIH example in fig. 7.5 and tab. 7.3: Clearly, SCF convergence becomes

---

view of implementation, where SCF convergence may well be spoiled by numerical artifacts as, e.g., discussed in sec. 7.2.3, cannot be assessed here in any detail.

worse with increasing basis set size. Whereas even the somewhat naïve Roothaan–Hall-type SCF algorithm converges faster for the smaller cc-pVDZ and aug-cc-pVDZ basis sets, the SCF ODA saves 7 iterations for the cc-pVTZ case and shows equally robust convergence behavior over the whole range of scenarios considered; contrasting, the Roothaan–Hall-type SCF algorithm fails to converge at all for basis sets larger than, and including, aug-cc-pVTZ.

**Tab. 7.3:** Numbers of SCF iterations  $\nu$  needed to converge  $E_{\text{SCF}}$  below  $\delta_E = 10^{-9}$  a.u. for the discussed TiH case, using different cc-pV $X$ Z and aug-cc-pV $X$ Z basis sets<sup>139,140</sup> with cardinality numbers  $X$ , for the Roothaan–Hall-type and the SCF ODA.

|                          | 2  | aug-2 | 3  | aug-3  | 4      | aug-4  |
|--------------------------|----|-------|----|--------|--------|--------|
| $\nu$ Roothaan–Hall-type | 15 | 18    | 33 | failed | failed | failed |
| $\nu$ ODA                | 23 | 22    | 25 | 24     | 27     | 26     |

## 7.4 Molecule-From-Atoms Initial Guess Densities

To further improve SCF convergence, particularly for early iterations in the SCF ODA loop, a “molecule-from-atoms” density (MFAD) initial guessing scheme has been implemented. As originally proposed by Almlöf et al.,<sup>38</sup> the 0-th molecular SCF density matrix  $\mathbf{D}^\mu = \mathbf{D}^0$  is constructed from SCF density matrices  $\mathbf{D}(A)$  for all atoms  $A$  of the molecular system of interest as

$$\mathbf{D}_{\gamma\gamma'}^0 = \mathbf{D}_{\gamma\gamma'}(1) \oplus \mathbf{D}_{\gamma\gamma'}(2) \oplus \cdots \oplus \mathbf{D}_{\gamma\gamma'}(A) \oplus \cdots \quad (7.35)$$

for  $\gamma, \gamma' \in \{\alpha, \beta\}$ .<sup>n</sup> Eq. 7-35 is similar to the ASA<sup>153</sup> and ADMA methods,<sup>154</sup> and is the first step of Jansík et al.’s multilevel strategy;<sup>155</sup> van Lenthe et al.’s<sup>156</sup> investigations demonstrate that MFAD are generally superior among typically available semiempirical and minimal basis initial guesses.

<sup>n</sup> The need for a block-wise definition of  $\mathbf{D}^0$  comes from the particular row and column ordering adopted in sec.s 3.2, 5.1.1, and 5.2.2. for hermitian time reversal-invariant matrix representations. With the  $\mathbb{H}^{n \times n} \rightarrow \mathbb{A}_n$  map  $\mathcal{M}$  of eq. 5-6, however, one can define  $\mathbf{D}^0 = \mathcal{M}(\bigoplus_A \mathcal{M}^{-1}\mathbf{D}(A))$ , which is employed in the implementation of eq. 7-35.

## Molecule-From-Atoms Initial Guess Densities

---

The following discussion is, again, restricted to the more general 2-component case. However, completely analog functionalities are provided for the spin-free 1-component HF SCF calculation scheme.

Technically, the top-level class `StandardMolecularHamiltonian2SpinorRepresentation_Container` has been given a method `provide_AtomicSCFDensityInitialGuess()` that returns a `HermitianTimeReversalInvariantMatrixRepresentation` object that is, then, employed as 0-th molecular SCF density matrix.

Central to `provide_AtomicSCFDensityInitialGuess()` is the auxiliary `CompactAtomic2cODASCFJob` class of a single template `CGTOBasis` parameter, matching the wrapping `StandardMolecularHamiltonian2SpinorRepresentation_Container`'s `CGTOBasis` argument. The single constructor from `const Molecule &` and two references to `const string`, defining the atom, the atomic CGTO basis set,<sup>o</sup> and the atomic occupation scheme, respectively, carries out an HF SCF calculation with an energy threshold of  $\delta_E = 10^{-8}$  a.u. The converged, quaternion-valued density matrix  $\mathcal{M}^{-1}\mathbf{D}$  is then retrieved via the `AtomicSCFDensityMatrix()` method.

Having wrapped the actual HF SCF calculation with `CompactAtomic2cODASCFJob`, `provide_AtomicSCFDensityInitialGuess()` runs over all `Atom` entries of the input `Molecule` and constructs a `std::map<Atom, CompactAtomic2cODASCFJob>` container: For every unique atom type, a corresponding single-entry `Molecule`, PP core size-specific occupation scheme `string`,<sup>p</sup> and basis set

<sup>o</sup> More precisely, the `const string &` defines the path, relative to the `$HOME/QOLBasis2` directory, of the respective CGTO basis set's XML file.

Note that both through the matching of `CompactAtomic2cODASCFJob`'s and `StandardMolecularHamiltonian2SpinorRepresentation_Container`'s `CGTOBasis` template argument, and through setting the `const string &` argument to `StandardMolecularHamiltonian2SpinorRepresentation_Container`'s basis set-defining attribute `_basis`, the basis set for the atomic calculation is restricted to the basis set employed for that atom in the subsequent molecular calculation.

<sup>p</sup> PP core size-dependent occupation schemes are provided by the auxiliary `AtomCore2OccupationSchemeMapper` class derived from `map<pair<string, unsigned int>, string>` that assigns element names (coded as `string`, and readily converted to `QOL::Molecule::TypeClass`) and PP core sizes to occupation schemes as discussed in sec. 7.2.3. Upon construction, `AtomCore2OccupationSchemeMapper` reads this assignment information from a C-array of

defining string is defined and handed to the respective CompactAtomic2cODA-SCFJob's constructor to run the calculation.

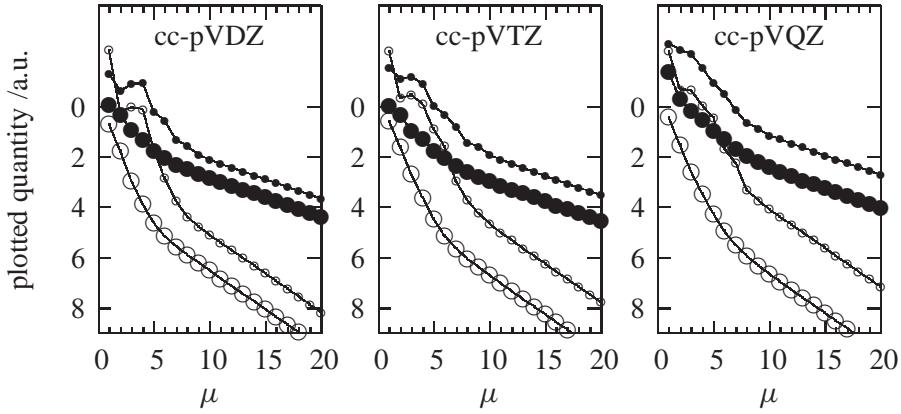
```
1 // provide_AtomicSCFDensityInitialGuess() const {
2
3 typedef std::map<Atom, CompactAtomic2cODASCFJob> mA2SCF;
4 mA2SCF AtomicSCF;
5 AtomCore2OccupationSchemeMapper OccMapper;
6
7 for(Molecule::const_iterator A = _molecule.begin();
8 A != _molecule.end(); ++A)
9 {
10 typename mA2SCF::const_iterator I = AtomicSCF.find(*A);
11 if(I == AtomicSCF.end())
12 {
13 const string Basis(_basis1.first);
14 const string Occ = OccMapper[make_pair(
15 A->typeClass().name(), A->PP().coreSize())];
16
17 AtomicSCF.insert(make_pair(*A,
18 CompactAtomic2cODASCFJob(*A, Basis, Occ)));
19 }
20 }
```

Then, iteration over all atomic centers  $\mathbf{r}_A$  of the CGTO basis set, and step-wise construction of the direct sum of eq. 7-35 from the atomic density matrix  $\mathbf{D}(A)$  of the atomic type at  $\mathbf{r}_A$ , gives rise to the same basis function ordering in  $\mathbf{D}^0$  and matrix representations constructed from the full molecular basis set.

The improvement of the SCF ODA convergence is illustrated in fig. 7.6 and tab. 7.4 for the TiH example already considered in sec. 7.3: For all cc-pV $X$ Z basis sets considered, the MFAD initial guess brings  $\|\mathbf{D}^1 - \mathbf{D}^0\|_2$  close to 1. Convergence is generally smooth and free of oscillations also for the very first iterations, and – for  $\delta_E = 10^{-9}$  a.u. – achieved in 19, 18, and 18 iterations as compared to 23, 25, and 27 for cardinal numbers  $X$  of 2, 3, and 4, respectively; analog conclusions apply to the considered aug-cc-pV $X$ Z basis set cases not displayed. Note, too, that the MFAD SCF ODA provides equally robust performance for all basis sets

---

AtomCoreString objects (declared **static** and) defined in StandardMolecularHamiltonian2SpinorRepresentation.Container.C.



**Fig. 7.6:** Comparison of core Hamiltonian and MFAD initial guess SCF ODA convergence: TIH at experimental  $d_{\text{TI}-\text{H}} = 3.5338$  a.u.,  $^{138}_{\text{81}}\text{Ti}$  small-core MC-DHF PP $^{131}$  with cc-pV $XZ$  basis,<sup>139,140</sup> for cardinality numbers  $X$  of 2, 3, and 4. Displayed are both  $-\lg |E^{\mu} - E^{\mu-1}|$  and  $-\lg \|D^{\mu} - D^{\mu-1}\|_2$  for core Hamiltonian ( $D^0 = \mathbf{0}_{2n}$ , small circles ○, ●) and MFAD initial guesses (big circles ○, ●), respectively.

considered – contrasting the core Hamiltonian initial guess cases.

**Tab. 7.4:** Comparison of core Hamiltonian and MFAD initial guess for the TIH example of fig. 7.6. Given are numbers of SCF iterations  $\nu$  needed to converge  $E_{\text{SCF}}$  below  $\delta_E = 10^{-9}$  a.u., 2-norms of differences of the first and last iteration's density matrix and  $D^0$  for MFAD initial guess matrices; the latter vanishes trivially for  $D^0 = \mathbf{0}_{2n}$ .

|                   | 2                              | aug-2 | 3     | aug-3 | 4      | aug-4  |
|-------------------|--------------------------------|-------|-------|-------|--------|--------|
|                   | MFAD initial guess             |       |       |       |        |        |
| $\nu$             | 19                             | 19    | 18    | 18    | 18     | 18     |
| $\ D^1 - D^0\ _2$ | 1.16                           | 1.37  | 1.01  | 2.27  | 22.69  | 11.97  |
|                   | core Hamiltonian initial guess |       |       |       |        |        |
| $\nu$             | 23                             | 22    | 25    | 24    | 27     | 26     |
| $\ D^1 - D^0\ _2$ | 20.94                          | 47.91 | 34.46 | 51.85 | 323.25 | 413.10 |

---

# 8

## 4-Index Integral Transformation and MP2

Almost all post-HF electronic structure theories that make use of expansion techniques in terms of 2-spinor Slater determinant many-electron functions refer to matrix elements of 1- and 2-electron operators in the “molecular 2-spinor”, i.e. a Fockian eigenbasis. For example, in the second-order Møller–Plesset perturbation theory energy expression

$$E_{\text{MP2}} = E_{\text{SCF}} + \frac{1}{4} \sum_{ijab} \frac{|g_{ijab} - g_{ijba}|^2}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}, \quad (8-1)$$

the indices  $i$ ,  $j$ ,  $a$ , and  $b$  refer to (occupied and virtual) 2-spinors  $|\psi\rangle$  defined by  $\langle\psi_i|\hat{f}\psi_j\rangle = \epsilon_j \delta_{ij}$ ; consequently,  $g_{ijab} = \langle\psi_i\psi_j|\hat{g}\psi_a\psi_b\rangle$ . However, the 2-spinors are generally different from the “atomic” CGTO basis functions in terms of which these are expanded for the discretization of the HF SCF equations.

Therefore, the first step in a post-HF calculation is the transformation of the “atomic 2-spinor basis”<sup>a</sup> 1-electron 2-index and 2-electron 4-index integrals to the molecular 2-spinor basis.

---

<sup>a</sup> From the expansion of the 2-spinors  $|\psi_i\rangle = \sum_p (C_{pi\alpha}|\phi_p\alpha\rangle + C_{pi\beta}|\phi_p\beta\rangle)$  in terms of “scalar”, i.e. not spinor-valued CGTO basis functions  $|\phi_p\rangle$ , the term “atomic 2-spinor basis” has to be used in a somewhat loose sense, meaning the basis of the  $n$  atom-centered CGTOs.

---

Based on the QOL 2-component HF SCF modules discussed in the preceding chapters, a 4-index integral transformation similar to the  $4n^5$  algorithm has been designed and implemented as a first step towards correlated calculation schemes. Integral indices are transformed pairwise for implicit spin integration as discussed in sec. 8.2.1. The naïve  $4n^5$  transformation algorithm of sec. 8.2.2 is formulated as a sequence of matrix–matrix multiplications, employing highly efficient BLAS3 routines and specialized matrix classes discussed in sec. 8.2.3. An exploratory application of 2-component MP2 theory to the Rn dimer is presented in sec. 8.3.

#### 4-Index Integrals: Index Notation, Restriction, and Symmetry

The connection of the atomic and molecular 2-spinor basis (AS and MS basis) 4-index integrals is given by eq. 3-23, i.e.

$$g_{ijkl} = \sum_{p\gamma} \sum_{q\gamma'} \sum_{r\gamma''} \sum_{s\gamma'''} C_{pi\gamma}^* C_{qj\gamma'}^* C_{rk\gamma''} C_{sl\gamma'''} g_{prqs} \quad (8-2)$$

for  $p, q, r, s \in \{1, \dots, n\}$  and  $\gamma, \gamma', \gamma'', \gamma''' \in \{\alpha, \beta\}$ . As already defined in sec. 3.2, here and in the following  $p, q, r, s$ ; and  $i, j, k, l$  are general AS and MS indices, respectively.

Where necessary, “occupied” and “virtual” will be labeled  $i, j, k, l$ ; and  $a, b, c, d$ , respectively, spanning the occupied and virtual subspaces

$$\begin{aligned} \mathbb{O} &= \text{span}(|\psi_i\rangle, |\bar{\psi}_i\rangle)_{i \in O} \subseteq \text{span}(|\psi_i\rangle, |\bar{\psi}_i\rangle)_{v_i=2}, \\ \mathbb{V} &= \text{span}(|\psi_a\rangle, |\bar{\psi}_a\rangle)_{a \in V} \subseteq \text{span}(|\psi_a\rangle, |\bar{\psi}_a\rangle)_{v_a=0}. \end{aligned} \quad (8-3)$$

Note that, for almost all post-HF calculations, typically only a subspace of  $\mathbb{O} \oplus \mathbb{V}$  is considered, i.e.  $t_O$  occupied and  $t_V$  virtual 2-spinor Kramers pairs of are discarded. For the ease of notation,  $\mathbb{O}$  and  $\mathbb{V}$  are understood to always exclude these  $2t_O + 2t_V$  2-spinors, and to be defined by the occupied and virtual index sets  $O := \{t_O + 1, \dots, N\}$  and  $V := \{N + 1, \dots, n - t_V\}$ , i.e. by the equality in 8-3.<sup>b</sup>

---

<sup>b</sup> In a small numbers of cases as, e.g., the general  $E_{\text{MP2}}$  expression of eq. 8-1, no particular reference to a Kramers-restricted calculation scheme is made, and  $\mathbb{O}$  and  $\mathbb{V}$  are understood accordingly as, e.g.,  $\mathbb{O} = \text{span}(|\psi_i\rangle)_{i \in \{2t_O+1, \dots, 2N\}} \subseteq \text{span}(|\psi_i\rangle)_{v_i=1}$ . However, no different notations will be employed, as the precise meaning of, e.g.,  $O$  will always be clear from the context.

Note, too, that for real-valued CGTO basis functions  $|\phi_p\rangle$  the AS basis integrals  $g_{prqs}$  are real, but generally  $g_{ijkl} \notin \mathbb{R}$ . Therefore, fewer symmetry relations, i.e.

$$g_{ijkl} = g_{jilk} = g_{klij}^* = g_{lkji}^*, \quad (8-4)$$

apply to the  $g_{ijkl}$  as compared to the  $g_{pqrs}$ . However, for a time reversal-invariant 2-spinor basis  $\mathcal{D} = (|\psi_i\rangle, |\bar{\psi}_i\rangle)_i$ , one has from eq.s 2-15, 2-17, and 2-21 the additional relations

$$g_{ijkl} = g_{il\bar{k}j}^*, \quad (8-5)$$

$$g_{i\bar{j}\bar{k}l} = -g_{ilkj}^*, \quad (8-6)$$

where the barred index  $\bar{i}$  is understood to label the  $i$ -th time reversal-transformed 2-spinor, i.e.  $g_{il\bar{k}j}^* = \langle \bar{\psi}_k \psi_j | \hat{g} \bar{\psi}_i \psi_l \rangle$ ; clearly,  $g_{ijkl} = g_{ij\bar{k}\bar{l}}^*$ .

## 8.1 Preliminary Considerations

The integral transformation of eq. 8-2 is typically implemented employing the so-called  $4n^5$  or “successive transformation” algorithm<sup>157</sup> Considering the simpler spin-free 1-component case of eq. 8-2, i.e. transforming to real-valued  $g_{ijkl}$  and dropping spin indices, the idea is to carry out the summation in four steps, re-writing

$$g_{ijkl} = \sum_p C_{ip} \sum_q C_{jq} \sum_r C_{rk} \sum_s C_{sl} g_{pqrs}. \quad (8-7)$$

For an  $\mathbb{R}^{n \times n \times n \times n} \rightarrow \mathbb{R}^{n \times n \times n \times n}$  transformation, the operation count scales as ca.  $4n^5$  with  $n$ : In the innermost and second-innermost sums  $\sum_s C_{sl} g_{pqrs} = g_{pqrl}$  and  $\sum_r C_{rk} g_{pqrl} = g_{pqkl}$  one has  $n$  multiplications (and additions) for all  $n^2$  ( $r, l$ ) and  $n^2$  ( $k, l$ ) index pairs, respectively, i.e.  $n \cdot n^2 + n \cdot n^2 = 2n^3$  operations; but this has to be done for all  $n^2$  ( $r, s$ ) index pairs, thus involving  $n^2 \cdot 2n^3$  operations. Computation of the two outermost sums via the same process gives thus a total operation count of  $4n^5$ .<sup>c</sup>

<sup>c</sup> For  $\mathbb{R}^{n \times n \times n \times n} \rightarrow \mathbb{R}^{m \times m \times m \times m}$  transformations to truncated Fockian eigenbases of dimensions  $m = n - t_O - t_V \leq n$ , the operation count is  $n^4 m + n^3 m^2 + n^2 m^3 + nm^4$ , i.e. approximately  $n^4 m$  for  $n \gg m$ . Considerations of index symmetry<sup>158–160</sup> can further reduce the computational effort.

Eq. 8-7 can be written as a sequence of four matrix–matrix multiplications and re-ordering steps: For example, the innermost sum over  $s$  can be written as

$$\sum_s C_{sl} g_{pqrs} = \sum_s g_{prqs} C_{ls} =: \sum_s I_{\tau_n(p,q,r)s} C_{sl} = (\mathbf{IC})_{\tau_n(p,q,r)l}, \quad (8-8)$$

employing the  $\mathbb{R}^{n^3 \times n}$  4-index integral matrix  $\mathbf{I}$ ;

$$\tau_n: (p, q, r) \mapsto \tau_n(p, q, r) = n^2 p + nq + r \quad (8-9)$$

is a map from the first three of the four 4-index integral indices to the  $\mathbf{I}$  matrix's row index. Then, after re-sorting the  $\mathbf{IC}$  matrix's elements as  $(\mathbf{IC})_{\tau_n(p,q,r)l} \mapsto (\mathbf{IC})_{\tau_n(l,p,q)r}$ , the second-innermost sum over  $r$  can be computed in exactly the same fashion, etc.

Whereas this key point, i.e. the step-wise computation of the four sums of eq. 8-7 as matrix–matrix multiplications, is also central to the implemented 2-component transformation algorithm, a somewhat different procedure has to be adopted from the 2-spinor nature of the 1-electron functions and the anti-unitarity of the time reversal operator.

## 8.2 Integral Transformation to Fockian Eigenbases

### 8.2.1 Broken Spin Symmetry: Index Pair Transformation

In the general 2-component case, the transformation of AS to MS basis 4-index integrals  $g_{ijkl}$  cannot be performed as a straightforward sequence of four matrix–matrix multiplications from the breaking of spin symmetry: Expanding a 4-index integral  $g_{ijkl}$  in terms of the  $i$ -th,  $j$ -th,  $k$ -th, and  $l$ -th 2-spinors' spin components gives

$$g_{ijkl} = g_{i_\alpha j_\alpha k_\alpha l_\alpha} + g_{i_\alpha j_\beta k_\alpha l_\beta} + g_{i_\beta j_\alpha k_\beta l_\alpha} + g_{i_\beta j_\beta k_\beta l_\beta}, \quad (8-10)$$

using the notation  $g_{i_\gamma j_\gamma' k_\gamma'' l_\gamma''' := g_{ijkl\gamma\gamma'\gamma''\gamma'''}}$  for the clarity of presentation.

However, eq. 8-10 follows from spin orthogonality and the spin-independence of  $\hat{g}$ , whereas the individual spatial integrals  $g_{i_\gamma j_\gamma' k_\gamma'' l_\gamma'''}$  generally do not vanish for arbitrary spin index combinations

Therefore, a 4-index integral transformation similar to the  $4n^5$  algorithm of sec. 8.1 must be carried out in two steps, transforming the electron-1 and -2 indices pairwise in each such step.

First of all, the 2-spinor expansion coefficient matrix  $\mathbf{C} \in \mathbb{A}_n$  is partitioned in four  $n \times n$  block matrices as

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_\alpha & \bar{\mathbf{C}}_\alpha \\ \mathbf{C}_\beta & \bar{\mathbf{C}}_\beta \end{pmatrix} := \left( \begin{array}{ccc|ccc} C_{1\alpha 1} & \cdots & C_{1\alpha n} & -C_{1\beta 1}^* & \cdots & -C_{1\alpha n}^* \\ \vdots & & \vdots & \vdots & & \vdots \\ C_{n\alpha 1} & \cdots & C_{n\alpha n} & -C_{n\beta 1}^* & \cdots & -C_{n\alpha n}^* \\ \hline C_{1\beta 1} & \cdots & C_{1\beta n} & C_{1\alpha 1}^* & \cdots & C_{1\alpha n}^* \\ \vdots & & \vdots & \vdots & & \vdots \\ C_{n\beta 1} & \cdots & C_{n\beta n} & C_{n\alpha 1}^* & \cdots & C_{n\alpha n}^* \end{array} \right) \quad (8-11)$$

i.e. in the time-forward and -reverse 2-spinor vector representations'  $\alpha$ - and  $\beta$ -spin component matrices  $\mathbf{C}_\alpha$ ,  $\bar{\mathbf{C}}_\alpha = -\mathbf{C}_\beta^*$ ,  $\mathbf{C}_\beta$ , and  $\bar{\mathbf{C}}_\beta = \mathbf{C}_\alpha^*$ , respectively.

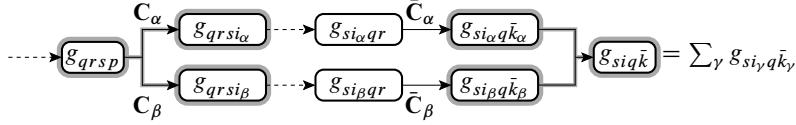
Transformation of the electron-1 indices  $p$  and  $r$  is done by separate multiplication of  $\mathbf{I} \in \mathbb{R}^{n^3 \times n}$  by, e.g.,  $\mathbf{C}_\alpha$  and  $\mathbf{C}_\beta$ , obtaining the intermediate  $\mathbb{C}^{n^3 \times n}$  matrices with elements  $(\mathbf{IC}_\alpha)_{\tau_n(q, r, s)i_\alpha} = g_{qrsi_\alpha}$  and  $(\mathbf{IC}_\beta)_{\tau_n(q, r, s)i_\beta} = g_{qrsi_\beta}$ , respectively. Then, re-ordering the intermediate matrices as  $(\mathbf{IC}_\gamma)_{\tau_n(q, r, s)i_\gamma} \mapsto (\mathbf{IC}_\gamma)_{\tau_n(s, i_\gamma, q)r}$ , is followed by respective multiplication with, e.g., the time-reverse matrices  $\bar{\mathbf{C}}_\alpha$  and  $\bar{\mathbf{C}}_\beta$ , and addition to the intermediate matrix with transformed electron-1 indices, i.e. with elements

$$g_{siq\bar{k}} = g_{si_\alpha q\bar{k}_\alpha} + g_{si_\beta q\bar{k}_\beta}.$$

Transformation of the electron-2 indices proceeds in exactly the same fashion, re-ordering of the intermediate matrix understood. The index pair transformation steps are illustrated in fig. 8.1.

Within a given pair transformation step, arbitrary truncations of the  $\mathbb{O} \oplus \mathbb{V}$  basis transformed to are readily achieved by employing  $n \times m$  sub-matrices of  $\mathbf{C}_\gamma$  for multiplication, i.e. by discarding the first  $t_O$  and the last  $t_V$  columns such that  $m = n - t_O - t_V$ . Then, the dimensions of the intermediate matrices change with every multiplication and re-ordering step. Consequently, the tensor-to-matrix index map  $\tau_n$  of eq. 8-9 has to be generalized to

$$\tau_{\mathbf{d}}: (p, q, r) \mapsto \tau_{\mathbf{d}}(p, q, r) := n_q n_r p + n_r q + r \quad (8-12)$$



**Fig. 8.1:** Illustration of a single pair transformation step: 4-index integral matrices are represented in terms of their 4-index integral matrix elements, e.g.,  $(\mathbf{IC}_\gamma)_{\tau_\mathbf{d}(q,r,s)i_\gamma} = g_{qrsi_\gamma}$ . Solid and dashed box-connecting arrows indicate matrix–matrix multiplication and index re-ordering steps, respectively.

and, thus, depends on all four indices’ domains collected in its vector index  $\mathbf{d} = (n_p, n_q, n_r, n_s)$ .

From similar arguments as given in sec. 8.1, the operation count for the first pair transformation can be estimated as  $8n^8m + 8n^3m + 8n^3m^2 + 8n^2m^2$ , including the  $n^3m$  and  $n^2m^2$  re-ordering steps and a factor of 4 for complex-valued arithmetic. Accordingly, the second pair transformation goes as  $8n^2m^3 + 8nm^3 + 8nm^4 + 8m^4$ .

## 8.2.2 A Naïve 4-Index Integral Transformation Algorithm

The pairwise index transformation step discussed in the preceding section is the basic building block of both the design and the implementation of the `QOL::SpinorTransform_MPPT` namespace’s class structure. Whereas these modules – as discussed in sec. 8.2.3 – provide a certain degree of flexibility for the implementation of more specialized transformation schemes, the discussion given here focuses on a general, “naïve” algorithm to compute a subset of MS basis 4-index integrals from that the complete set can be generated by symmetry relations among these integrals.

Considering time reversal-symmetry only, from a given AS basis 4-index integral  $g_{pqrs}$ , a total of 16 possible MS basis integral types can be – and generally have to be – generated, i.e.

$$g_{ijkl}, g_{ij\bar{k}\bar{l}}, g_{i\bar{j}kl}, g_{i\bar{j}\bar{k}\bar{l}}, g_{\bar{i}jkl}, g_{i\bar{j}kl}, g_{ij\bar{k}l}, g_{ij\bar{k}\bar{l}}, \\ g_{i\bar{j}\bar{k}\bar{l}}, g_{\bar{i}jk\bar{l}}, g_{\bar{i}j\bar{k}l}, g_{\bar{i}\bar{j}kl}, g_{\bar{i}\bar{j}k\bar{l}}, g_{ij\bar{k}\bar{l}}, \text{ and } g_{i\bar{j}\bar{k}l}.$$

However, because of the anti-linearity of the time reversal operator one generally

cannot, from a given  $g_{ijkl}$ , generate all other 15 integrals. In fact, using eq.s 8-4, 8-5, and 8-6, the 16 integral types can be collected in four “families” as

$$g_{ijkl} = g_{il\bar{k}j}^* = g_{k\bar{j}i\bar{l}}^* = g_{\bar{k}\bar{l}\bar{i}\bar{j}} \quad \text{family 1; \quad (8-13)}$$

$$\begin{aligned} -g_{jkli}^* &= -g_{\bar{k}\bar{j}il}^* = g_{jl\bar{k}} = g_{ij\bar{k}\bar{l}} \\ = -g_{lk\bar{j}i}^* &= -g_{\bar{k}\bar{l}ij}^* = g_{l\bar{i}\bar{j}\bar{k}}^* = g_{\bar{i}\bar{l}\bar{k}\bar{j}}^* \end{aligned} \quad \left. \right\} \quad \text{family 2; \quad (8-14)}$$

$$g_{\bar{k}\bar{l}\bar{i}\bar{j}}^* = g_{ij\bar{k}\bar{l}} \quad \text{family 3; \quad (8-15)}$$

$$g_{\bar{i}\bar{j}\bar{k}\bar{l}}^* = g_{ij\bar{k}\bar{l}} \quad \text{family 4, \quad (8-16)}$$

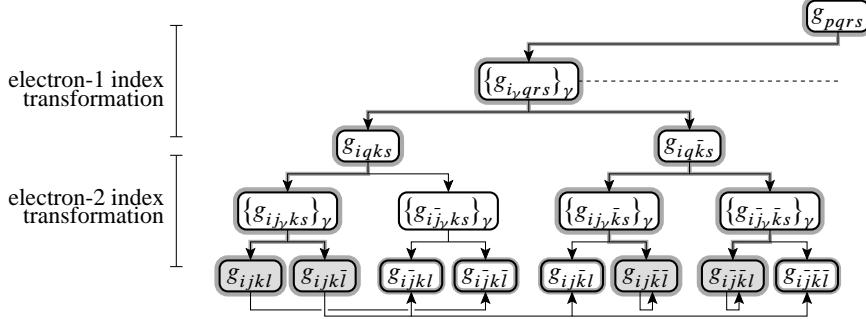
such that a given family’s members can be generated from each other by hermitian conjugation, electron–electron interchange, and time reversal transformation, or any combinations thereof.<sup>d</sup> It is thus necessary, but typically sufficient to compute only four  $\mathbb{C}^{m^3 \times m}$  MS basis 4-index integral matrices, i.e. one for each integral family.

Whereas such four matrices are clearly distinct, some share a number of intermediates over the course of their computation from the AS basis 4-index integral matrix **I**. For example, the family 1 and family 2 integrals  $g_{ijkl}$  and  $g_{ijk\bar{l}}$  share the electron-1 indices-transformed intermediate  $g_{iqks}$  and, in fact, also the pair  $\{g_{qj_\gamma ks}\}_\gamma$ .

It is this observation that defines the 2-component 4-index integral transformation scheme illustrated in fig. 8.2. A total of 20 matrix–matrix multiplications is necessary to arrive at the four matrix representatives with elements  $g_{ijkl}$ ,  $g_{ijk\bar{l}}$ ,  $g_{ij\bar{k}\bar{l}}$ , and  $g_{ij\bar{k}\bar{l}}$ , because different intermediates  $\{g_{ij_\gamma \bar{k}s}\}_\gamma$  and  $\{g_{ij_\gamma \bar{k}s}\}_\gamma$  have to be computed for  $g_{ijk\bar{l}}$  and  $g_{ij\bar{k}\bar{l}}$ , respectively, in the two right branches of fig. 8.2; consequently, 10 re-sorting steps are needed.

---

<sup>d</sup> Note that the definition of the four integral families according to eq.s 8-13, 8-14, 8-15, and 8-16, allows time reversal transformation of only a single electron’s 2-spinors and, by eq.s 8-5 and 8-6, implies swapping the other electron’s 2-spinors among the anti-linear and linear arguments of the inner product. This can be too restrictive for specialized applications that employ transformations of the bra and ket 2-spinors to different subspaces of  $\mathbb{O} \oplus \mathbb{V}$  as briefly discussed in sec. 8.3.



**Fig. 8.2:** Illustration of the 2-component 4-index integral transformation algorithm: Matrices are represented in terms of their 4-index integral elements, e.g.,  $(\mathbf{IC}_\gamma)_{\tau_d(i_\gamma, q, r)s} = g_{i_\gamma qrs}$ , index re-orderings omitted for clarity. Only one half of the complete tree is shown; the second half is analog with the electron-1 bra index barred. In the present implementation, only the four gray-shaded matrix representatives are computed explicitly along the gray-shaded path. Different representatives of integral families are connected by arrows.

### 8.2.3 BLAS3 Multiplication-Driven Auxiliary and Matrix Classes

Central to the matrix–matrix multiplication-driven implementation of the 2-component 4-index integral transformation algorithm discussed above is, of course, efficient multiplication as, e.g., provided by the Level 3 BLAS CGEMM and ZGEMM (complex-valued single and double floating point precision) routines.

Beside the QOL::MatrixVector::Matrix class already discussed in sec. 4.3.1, the 2006 QOL implementation provides a second, to some extent different matrix class in the QOL::LAPACK\_BLAS namespace. Both classes’ interfaces are largely similar and provide analog access of matrix elements via two-index () operators, and basic matrix algebra operations. Without going into too much technical detail, however, two main differences are of importance for the present discussion: First, QOL::LAPACK\_BLAS::Matrix provides a global function

```

1 template <class T>
2 void mult_2ndTransposed(
3 const QOL::LAPACK_BLAS::Matrix<T> & A,
4 const QOL::LAPACK_BLAS::Matrix<T> & B,
5 QOL::LAPACK_BLAS::Matrix<T> & AB,
6 const T &);
```

for BLAS3-driven multiplication; `mult_2ndTransposed` takes the product matrix as third argument to bypass the need to return a copy of the product. Second, `Matrix` does not allow to choose different underlying evaluator–container classes for, e.g., only non-redundant matrix element memory allocation. Instead, `Matrix` employs a simple `boost::shared_ptr<std::vector<T>>` and a flat () operator only. Consequently, the `QOL::LAPACK_BLAS::Matrix` class cannot exploit matrix symmetry or internal structure.

For the matrix–matrix multiplication-driven transformation of real-valued AS to complex-valued MS basis 4-index integrals, an additional `conjugate()` method was added to `Matrix`. For both template arguments `std::complex<float>` and `std::complex<double>`, the LAPACK/BLAS CGEMM and ZGEMM subroutines have been interfaced as `mult_2ndTransposed` specializations for the respective cases.

The `LAPACK_BLAS_4IndexIntegralMatrix` class is, then, derived from this `QOL::LAPACK_BLAS::Matrix` by inheritance. As index mapping and re-ordering is assisted by the `ActiveIndex`, `Integral4Index_CyclicPermutation`, `Ten-`  
`sor2IndexMapper` auxiliary classes, these are briefly discussed first.

### Index Mapping and Re-Ordering Auxiliary Classes

It is clear from sec.s 8.2.1 and 8.2.2 that the mapping  $\tau_d$  of the four integral to the two matrix indices must change dynamically – both because of necessary matrix re-ordering steps and, in the general case of transformations to truncated 2-spinor bases, varying index ranges. Therefore, a small number of auxiliary classes have been provided to assist index operations by and on the top-level `LAPACK_BLAS_-4IndexIntegralMatrix` class. The key design ideas are, first, that all operations on the 4-index integrals' index 4-tupel are cyclic permutations and, second, every such permutation is defined by an “active” index to be transformed.

- `ActiveIndex` is a simple enumeration type i.e.

```
enum ActiveIndex { ket2, ket1, bra2, bra1 };
```

Note that `ket2`, `ket1`, `bra2`, and `bra1` are mapped to the integer values 0, 1, 2, and 3, respectively, by implicit type conversion.

- `Integral4Index_CyclicPermutation` tracks index 4-tupel ordering by its `int _P` attribute: It provides a single `int map2_mod4(const int & p)` method only, returning the rest of division by 4,<sup>e</sup> and `+=`, `-=`, `+`, and `-` operators for addition and subtraction modulo 4.

In this way `Integral4Index_CyclicPermutation` mimicks the (commutative) group structure of the cyclic permutations  $C_4$  by  $\{0, 1, 2, 3\}$  and addition modulo 4.<sup>f</sup> The connection to the four integral indices is made by `ActiveIndex`, i.e. the value of `_P` defines which integral index is the matrix column index and, thus, is to be transformed.

- The `Tensor2IndexMapper` class is derived from `std::vector<int>` and effectively provides the  $\tau_d$  mapping of eq. 8-12. Mapping of its four entries, corresponding to the index ranges, is established by its `Integral4Index_CyclicPermutation` attribute: If this is the identity permutation, the vector's first, second, third, and fourth entry corresponds to the electron-1 bra, electron-2 bra, electron-1 ket, and electron-2 ket index's maximum value, respectively.

Index range re-ordering is enabled by the `void setActiveIndex(const ActiveIndex & A)` method: `A` is – via implicit type conversion – “added” to `Integral4Index_CyclicPermutation`, which effectively computes the cyclic permutation required to go from the current to the specified index range ordering, and re-orders the vector entries accordingly.

### BLAS3 Multiplication-Driven 4-Index Integral Matrix Classes

The top-level `LAPACK_BLAS_4IndexIntegralMatrix` class is a single-parameter template of argument `T`, assumed to be of `std::complex<float>` or `std::complex<double>` types. It is a `QOL::LAPACK_BLAS::Matrix` by inheritance

---

<sup>e</sup> C++ and, e.g., FORTRAN provide the ‘symmetric’ modulo function  $a \bmod n := a - n \cdot \text{int } a/n$ , with  $\text{int } a/n$  the integer part of  $a/n$ . However, for  $m \in \mathbb{Z}$ , generally  $a \bmod n \neq (a + m \cdot n) \bmod n$  for this function. Therefore, `map2_mod4` has been implemented to return  $a - n \cdot \lfloor a/n \rfloor$  with the desired property.

<sup>f</sup> That is,  $C_4$  is mimicked by the quotient group  $\mathbb{Z}/4\mathbb{Z}$ , i.e. the group of the rest classes modulo 4 with addition. In the book of Fischer<sup>129</sup> cyclic groups  $C_n$  are defined in terms of  $\mathbb{Z}/n\mathbb{Z}$ .

and, thus, makes use of all of the base classes' methods and attributes, particularly of the BLAS3 CGEMM- and ZGEMM-driven `mult_2ndTransposed` function. As already pointed out, index operations are assisted by an additional `Tensor2IndexMapper` attribute `_m`, enabling the definition of a four-index operator

```
1 const T & operator () (int i, int j, int k, int l) const {
2 return (*this)(_m[1]*_m[2]*i + _m[2]*j + k, l);
3 }
```

in terms of `Tensor2IndexMapper`'s vector entries and the base classes' two-index `()` operator; this is precisely the index map

$$(i, j, k, l) \mapsto (\tau_d(i, j, k), l) = (n_j n_k i + n_k j + k, l)$$

induced by eq. 8-12.

Index re-ordering operations have been wrapped in a `void setActiveIndex(const ActiveIndex & A)` method: If called, a new `LAPACK_BLAS_4IndexIntegralMatrix` object is constructed from `A`, setting the respective row and column dimensions from the integral index ranges. Then, the entries of the `LAPACK_BLAS_4IndexIntegralMatrix` to be re-sorted are written to the new object as defined by the cyclic permutation connecting the two index orderings, and the latter is assigned to the former.

The `SpinComponentPair_4IndexIntegralMatrix` class has been implemented only to assist organization of the complete 2-component 4-index integral transformation procedure in terms of pair transformation steps, and does not provide any special functionality itself. Put briefly, `SpinComponentPair_4IndexIntegralMatrix` inherits from `std::pair` of `LAPACK_BLAS_4IndexIntegralMatrix` and has a constructor from a `const LAPACK_BLAS_4IndexIntegralMatrix &` and two `const Matrix &` arguments, mimicking the construction of two spin index-labeled intermediate matrices, e.g.  $\mathbf{IC}_\gamma$  with elements  $(\mathbf{IC}_\gamma)_{\tau_d(q,r,s)i_\gamma} = g_{qrsi_\gamma}$  as illustrated in fig. 8.1.

In turn, the `LAPACK_BLAS_4IndexIntegralMatrix` class has been equipped by a constructor from a `const SpinComponentPair_4IndexIntegralMatrix &` and, similarly, two `const Matrix &` arguments to wrap the pair transformation step's second part, i.e. separate multiplication of the spin index-labeled intermediate matrices by coefficient matrices from the right, and addition.

With this class structure established, the electron-1 index transformation of fig. 8.2, i.e. the transformation from the AS basis 4-index integrals  $g_{pqrs}$  to intermediates  $g_{iqks}$  and  $g_{iq\bar{k}s}$ , can, for example, be written as

```
1 // typedef LAPACK_BLAS_4IndexIntegralMatrix<complex<
2 // double>> I4Matrix;
3 // typedef SpinComponentPair_4IndexIntegralMatrix<complex<
4 // double>> I4Matrix_Pair;
5
6 // I_pqrs CONTAINS ATOMIC 2-SPINOR BASIS 4-INDEX INTEGRALS
7 // WITH bra1 AS "ACTIVE" INDEX
8
9 I4Matrix_Pair I_iqrs(I_pqrs, fwdC_alpha, fwdC_beta);
10 I_iqrs.setActiveIndex(ket1);
11
12 I4Matrix I_iqks(I_iqrs, fwdC_alpha, fwdC_beta);
13 I4Matrix I_iqKs(I_iqrs, revC_alpha, revC_beta);
```

In a very similar way, also transformation schemes different from that discussed in sec. 8.2.2 can easily be assembled within the same modular structure, including, for example, computation of different or all integral family representatives, transformation of bra and ket indices to different subspaces of  $\mathbb{O} \oplus \mathbb{V}$ , etc. However, none of these will be considered in any more detail at this point. Instead, this discussion is closed with the presentation of preliminary applications at 2-component Møller–Plesset perturbation theory level in the next section.

### 8.3 Kramers-Restricted Møller–Plesset Perturbation Theory

Having the AS basis 4-index integrals  $g_{pqrs} \in \mathcal{G}_n$  transformed to the MS basis, it is straightforward to evaluate the MP2 energy expression of eq. 8-1.

Within the 2-component 4-index integral transformation scheme of sec. 8.2.2 that, for every integral family of eq.s 8-13, 8-14, 8-15, and 8-16, computes only one representative,  $E_{MP2}$  is cast to a form involving only these representative integrals as follows:

Expanding the square modulus and using hermiticity and electron–electron

symmetry of the  $g_{ijab}$ , one obtains

$$\frac{1}{4} \sum_{ijab} \frac{|g_{ijab} - g_{ijba}|^2}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} = \frac{1}{4} \sum_{ijab} \frac{g_{ijab}g_{abij} - g_{ijab}g_{abji}}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \quad (8-17)$$

for unrestricted summations over all  $2N - t_O$  occupied 2-spinor indices  $i, j$ , and all  $2n - t_V - 2N - t_O$ . Defining the complex-valued MP2 “amplitudes”

$$T_{ijab} := \epsilon_{ijab} \cdot g_{ijab}^* := \frac{g_{ijab}^*}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}, \quad (8-18)$$

eq. 8-17 is written in terms of Kramers pairs of 2-spinors as

$$\begin{aligned} E_{\text{MP2}} = & \frac{1}{2} \sum_{ijab} (T_{ijab} - T_{ijba})g_{ijab} + \frac{1}{2} \sum_{ij\bar{a}} (T_{ij\bar{a}} - T_{ij\bar{b}})g_{ij\bar{a}} \\ & + \frac{1}{2} \sum_{ij\bar{a}} (T_{ij\bar{a}} - T_{ij\bar{b}})g_{ij\bar{a}} + \frac{1}{2} \sum_{ij\bar{a}} (T_{ij\bar{a}} - T_{ij\bar{b}})g_{ij\bar{a}} \\ & + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} \\ & + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} \\ & + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} \\ & + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} \\ & + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} \\ & + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} + \frac{1}{2} \sum_{i\bar{j}ab} (T_{i\bar{j}ab} - T_{i\bar{j}ba})g_{i\bar{j}ab} \end{aligned}$$

Then, from eq.s 8-4, 8-5, and 8-6, every of the 16 terms can be expressed in terms of integrals of the types  $g_{ijab}$ ,  $g_{ijaa}$ ,  $g_{ij\bar{a}\bar{b}}$ , and  $g_{i\bar{j}\bar{a}\bar{b}}$  as collected in tab. 8.1; note that the  $\epsilon_{ijab}$  are identical for all combinations of barred and unbarred indices.

Technically, the QOL::SpinorTransform\_MPPT namespace provides a \_2c-MPPTnEvaluator template class of argument T that effectively evaluates  $E_{\text{MP2}}$  via tab. 8.1, i.e., for all  $(i, j, a, b) \in O \times O \times V \times V$ , adds all 16 rows of the second column of tab. 8.1. All the necessary information is provided to the single constructor from four **const** LAPACK\_BLAS\_4IndexIntegralMatrix & arguments, corresponding to the matrices with elements  $g_{ijab}$ ,  $g_{ijaa}$ ,  $g_{ij\bar{a}\bar{b}}$ , and  $g_{i\bar{j}\bar{a}\bar{b}}$ ; a **const** std::vector<typename T::value\_type> & argument holding the 2-spinor energies  $\epsilon_i$ ; and a **const unsigned int** &, corresponding to the num-

**Tab. 8.1:** Re-writing the Kramers-restricted MP2 energy in terms of the representative integrals computed within the 2-component 4-index integral transformation algorithm of sec. 8.2.2. The  $\blacktriangleleft$ -labeled contributions to  $E_{\text{MP2}}$  refer to integrals from different integral families.

| $T_{ijab} = \epsilon_{ijab} \cdot g_{ijab}^*$              | $= \frac{g_{ijab}^*}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}$                            |
|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| $(T_{ijab} - T_{ijba})g_{ijab}$                            | $\mapsto \epsilon_{ijab} (+g_{ijab}^* - g_{ijba}^*)g_{ijab}$                                        |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{ij\bar{a}}$             | $\mapsto \epsilon_{ijab} (+g_{jiba}^* - g_{ij\bar{b}}^*)g_{jiba}$                                   |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{ij\bar{b}}$             | $\mapsto \epsilon_{ijab} (+g_{ij\bar{a}}^* - g_{jiba}^*)g_{ijab}$                                   |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{ij\bar{a}}$             | $\mapsto \epsilon_{ijab} (+g_{ij\bar{a}}^* - g_{ij\bar{b}}^*)g_{ij\bar{a}}$                         |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{ij\bar{b}}$             | $\mapsto \epsilon_{ijab} (+g_{ij\bar{a}}^* - g_{ij\bar{b}}^*)g_{ij\bar{b}}$                         |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{a}}$ | $\mapsto -\epsilon_{ijab} (-g_{ija\bar{b}} + g_{jia\bar{b}})g_{ij\bar{a}}$                          |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{b}}$ | $\mapsto -\epsilon_{ijab} (-g_{jiba} + g_{ij\bar{b}})g_{jiba}$                                      |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{a}}$ | $\mapsto \epsilon_{ijab} (+g_{ij\bar{a}} - g_{ij\bar{b}})g_{ijab}$                                  |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{b}}$ | $\mapsto -\epsilon_{ijab} (-g_{bij\bar{a}} + g_{aij\bar{b}})g_{bij\bar{a}}$                         |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{a}}$ | $\mapsto \epsilon_{ijab} (+g_{ibaj} - g_{i\bar{j}\bar{b}})g_{ibaj} \blacktriangleleft$              |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{b}}$ | $\mapsto \epsilon_{ijab} (+g_{ij\bar{a}} - g_{i\bar{a}b\bar{j}})g_{ij\bar{a}} \blacktriangleleft$   |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{a}}$ | $\mapsto -\epsilon_{ijab} (-g_{abi\bar{j}} + g_{bai\bar{j}})g_{abi\bar{j}}$                         |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{b}}$ | $\mapsto -\epsilon_{ijab} (-g_{aj\bar{i}} + g_{bj\bar{a}})g_{aj\bar{i}}$                            |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{a}}$ | $\mapsto \epsilon_{ijab} (+g_{ij\bar{a}}^* - g_{i\bar{a}b\bar{j}})g_{ij\bar{a}} \blacktriangleleft$ |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{b}}$ | $\mapsto \epsilon_{ijab} (+g_{ibaj}^* - g_{i\bar{j}\bar{b}})g_{ibaj} \blacktriangleleft$            |
| $(T_{ij\bar{a}} - T_{ij\bar{b}})g_{\bar{i}\bar{j}\bar{a}}$ | $\mapsto -\epsilon_{ijab} (-g_{baji} + g_{abi\bar{j}})g_{baji}$                                     |

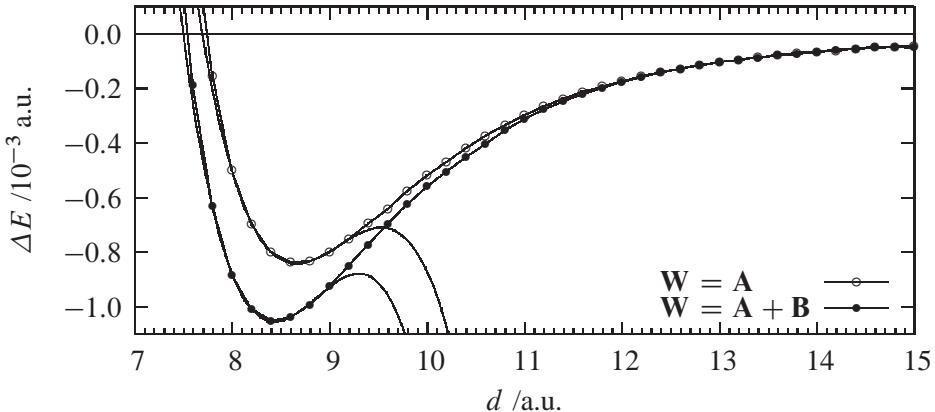
ber  $2N - t_O$  of electrons.<sup>g</sup>

Tab. 8.2 presents equilibrium bond lengths  $r_e$ , dissociation energies  $D_e$ , and har-

<sup>g</sup> Implementation of a member function as, e.g., “get\_EMP2Contribution()”, for the LAPACK-BLAS\_4IndexIntegralMatrix class is spoiled by the fact that, as indicated in tab. 8.1, some of the contributions to  $E_{\text{MP2}}$  involve MS basis 4-index integrals from different families (if these are defined according eq.s 8-13, 8-14, 8-15, and 8-16).

**Tab. 8.2:** Equilibrium distances  $d_e$ , dissociation energies  $D_e$ , and harmonic frequencies  $\omega_e$  for the  $^{222}_{86}\text{Rn}$  dimer,  $^{86}\text{Rn}$  small-core MC-DHF PP and cc-pVTZ basis set.<sup>130</sup> computed from the interpolation polynomials of fig. 8.3.

|                                               | $d_e$  | $D_e$                  | $\omega_e$             |
|-----------------------------------------------|--------|------------------------|------------------------|
| MP2, $\mathbf{W} = \mathbf{A}$ ,              | 8.6581 | $0.8421 \cdot 10^{-3}$ | $5.0882 \cdot 10^{-5}$ |
| MP2, $\mathbf{W} = \mathbf{A} + \mathbf{B}$ , | 8.4339 | $1.0551 \cdot 10^{-3}$ | $5.9115 \cdot 10^{-5}$ |
| Pyykkö et al., <sup>161</sup> MP2             | 8.6265 | $1.2899 \cdot 10^{-3}$ | $7.9280 \cdot 10^{-5}$ |
| Pyykkö et al., <sup>161</sup> CCSD(T)         | 8.7721 | $1.0106 \cdot 10^{-3}$ | $6.7434 \cdot 10^{-5}$ |



**Fig. 8.3:** Difference  $\Delta E$  of the  $\text{Rn}_2$  total MP2 energy  $E_{\text{MP2}}$  and the respective monomer energies as a function of interatomic separation  $d$ ,  $^{86}\text{Rn}$  small-core MC-DHF PP and cc-pVTZ basis set.<sup>130</sup> Plotted are pointwise computed  $\Delta E$  values and smooth interpolation polynomials fitted to the seven points near the minima.

monic frequencies  $\omega_e$  calculated for the  $^{222}_{86}\text{Rn}$  dimer at the MP2 level of theory,<sup>h</sup> using the small-core MC-DHF PP and cc-pVTZ basis of Peterson et al.,<sup>130</sup> both without and with the PP's spin-orbit part. Clearly, spin-orbit effects are large and

<sup>h</sup> The underlying transformation to the 2-spinor basis has not been truncated, i.e. all 2-spinors have been correlated.  $d_e$ ,  $D_e$ , and  $\omega_e$  have been calculated from interpolation polynomials  $A + B(d - d_e)^2 + C(d - d_e)^3$  fitted to seven points near the  $\Delta E$  minima of fig. 8.3. Precisely,  $D_e = A$  and  $\omega_e = \sqrt{2B/m}$ , with  $m = 4.0470 \cdot 10^5$  a.u. for  $^{222}_{86}\text{Rn}$ .<sup>162</sup>

found to shorten  $d_e$  by  $0.2242 \text{ a.u.}$  or  $11.9 \text{ pm}$ .  $\text{Rn}_2$  is bound by  $1.0551 \cdot 10^{-3} \text{ a.u.}$  at the 2-component level, which is more stable by  $0.2130 \cdot 10^{-3} \text{ a.u.}$  or  $5.80 \text{ meV}$  (ca. 25 %, without BSSE correction) than compared to the spin-free, 1-component calculation scheme.

Direct comparison with the large-core spin-free PP MP2, CCSD, and CCSD(T) calculations of Runeberg and Pyykkö<sup>161</sup> that include SO-CISD corrections, is difficult because of the different calculation schemes. However, whereas the exploratory results of this work should not be considered too conclusive, it cannot be judged which values are more accurate because of, e.g., the rather restrictive 4-index integral transformation truncation scheme employed for the SO-CISD corrections of Runeberg and Pyykkö.<sup>161</sup>

---

# 9

## Summary and Outlook

At the time of writing, i.e. in spring 2010, the research goals stated in sec. 4.2.1 have been accomplished: The Quantum Object Library’s 1-component spin-free Hartree–Fock SCF parts have been modified and extended to a set of 1-component spin-restricted all-electron and spin-free PP, and 2-component Kramers-restricted spin-orbit PP HF SCF programs. All are compatible with, and integrated in, both the established QOL structure and those parts subject of ongoing development.

The detailed discussions given in ch.s 5, 6, 7, and 8 are summarized as follows:

### ■ Ch. 5: Matrix Algebra:

The QOL matrix and matrix representation algebra modules have been supplemented by two class hierarchies corresponding to the `HermitianMatrixRepresentation` and `HermitianTimeReversalInvariantMatrixRepresentation` classes. The separation of basis functions’ domains and `UnitarySpace` scalar product codomains introduced complex-valued algebra on the matrix representation level, from basic arithmetic operations to equation solving, without substantial changes to the established class interfaces.<sup>a</sup> The inheritance tree of the `HermitianTimeReversalInvariant`

---

<sup>a</sup> The single relevant exception is that, now, a `const UnitarySpace &`-type argument has been passed to the matrix representation class constructors from the need to decouple algebraic “representation” and iteration-related “index” spaces, as discussed in sec. 5.2.1.

---

MatrixRepresentation hierarchy includes an auxiliary HermitianQuaternionMatrixRepresentation class for future interfaces to quaternion algebra modules.

## ■ Ch. 6: 2-Component Pseudopotentials:

PPs have been introduced in the “pseudo-atom” and “-molecule” framework of sec. 6.1, i.e. assigning a (generally vanishing) SemilocalAtomicPseudopotential to all Atom objects. No modifications of the Atom and Molecule class interfaces have been made; the all-electron use-case is recovered if no PP definitions are given. PP parameter organization in XML files allows easy data communication and checks for syntactic and semantic consistency.

PP integrals over CGTO basis functions are computed using the interfaced ARGOS PP integral subroutines<sup>31–35</sup> of Pitzer et al. The McMDPseudopotential\_Evaluator class wraps ARGOS common block definition, subroutine calls, and integral retrieval from the C++ side. PP integrals are handled quaternion-valued to allow evaluation and communication of spin-free and spin-orbit contributions simultaneously, employing the QOL’s established iterator-evaluator structure.

## ■ Ch. 7: Self-Consistent 2-Spinor Fields:

The new functionalities have been combined to give rise to two parallel, essentially analog, i.e. 1-component and 2-component HF SCF calculation schemes. From the use-case point of view, both differ in rarely more than a number of type definitions as, e.g., `double` vs. `std::complex<double>`. Top-level classes as TimeReversalInvariantRoothaanHall2cSCF\_EigenSystem encapsulate equation solving and eigenvector processing, i.e.: Fock matrix diagonalization; eigenspace-wise Kramers-like orthogonalization, unitary symplectic transformation, and re-phasing to enforce time reversal invariance; occupation number-to-eigenvector assignment; and density matrix assembly.

The SCF Optimal Damping Algorithm<sup>36,37</sup> has been adopted to the 2-component setting and significantly improves, or actually enables, SCF convergence. Initial guess density matrices constructed as direct sums of atomic densities,<sup>38,156</sup> allowing fractional atomic occupation numbers, further im-

prove SCF performance.

■ **Ch. 8: 4-Index Integral Transformation and MP2:**

Transformation of 4-index integrals to the molecular 2-spinor basis has been implemented similar to the  $4n^5$  algorithm,<sup>157</sup> written as matrix–matrix multiplications. A special `LAPACK_BLAS_4IndexIntegralMatrix` class has been provided, calling BLAS3 subroutines for efficient multiplication. Auxiliary classes exploit the group structure of integral index permutations to keep track of index mappings and ranges generally changing with every transformation step. The provided class structure allows modular, intuitive implementations of general and special-purpose transformation algorithms. Exploratory 2-component MP2 calculations of the Rn<sub>2</sub> potential curve on the basis of a full 2-spinor space 4-index integral transformation, exploiting time reversal-invariance, prove the principle.

Before turning to a critical discussion of points of principal and future interest, it is noted that none of the issues addressed in this work – with the exception of the 4-index integral transformation to the Fockian eigenbasis presented in ch. 8 – is performance-critical. A given HF SCF calculation’s demand of computational resources, i.e. CPU time, is almost exclusively determined by 4-index integral evaluation; the same is true for memory requirements if integral-conventional schemes are considered.<sup>b</sup>

For example, the present ARGOS–QOL interface, including PP integral evaluation and communication from the QOL side, surely does not exploit the ARGOS subroutines’ full capability. At the time, PP integrals are calculated for every distinct pair of primitive CGTOs separately. Contrasting, ARGOS allows evaluation of PP integrals over all (symmetry-adapted and contracted) CGTO pairs that arise from all combinations of Cartesian monomials generated from the CGTO pair considered, in a single call. However, this is irrelevant for the overall performance, and it is expected that the current implementation will remain unchanged for the foreseeable future. A re-implementation is more likely to be addressed from

---

<sup>b</sup> However, note that, particularly in the 2- and 4-component framework, Fock matrix diagonalization has been observed to contribute significantly to the overall computational effort<sup>163,164</sup> for density fitting-driven integral-direct or -semi-direct HF or Kohn–Sham DFT SCF calculations.

---

the point of view of the current limitations to PP integrals for CGTOs with  $l < 5$  and  $l < 4$  for the PP  $\hat{A}$  and  $\hat{B}$  parts, respectively.

Considering the actual SCF algorithm, two main points might be addressed in the near future: First, for large systems, the  $\mathbf{F}$  matrix diagonalization step can become performance-relevant;<sup>163,165</sup> quaternion diagonalization techniques,<sup>111</sup> exploiting time reversal symmetry of the problem, can speed up this step by roughly a factor of 2,<sup>111c</sup> but it is expected that other, more critical issues will have to be addressed before. Second, combination of the SCF ODA with convergence acceleration methods as DIIS<sup>150</sup> or, better, EDIIS<sup>36,151,152</sup> is a logical next step towards improved SCF performance.

As already stated implicitly by the limited size of the exemplary calculations presented in ch.s 7 and 8, the most critical point from the point of view of the present implementation is the QOL's 4-index integral part.

At the moment, the QOL provides a naïve, integral-conventional framework only, and integrals are evaluated for every unique index combination separately; although exploratory implementations exist, pre-screening does not pay off in this setting. Clearly, this restricts the applicability of all QOL HF SCF programs both in terms of requirement of CPU time and memory.

It is therefore of pivotal importance to re-work and re-implement the QOL integral and, consequently, iterator–evaluator modules more efficiently. Preliminary, yet uncompleted experiments with code-generated modules in the Hanrath group are promising, but the code is not operational at this time. Any implementation of integral-direct SCF algorithms, relying on the code-generated modules, will surely have to be done along the line with considerations of integral pre-screening techniques and integral-direct  $\mathbf{D}$  matrix assembly<sup>d</sup> – the latter having to be ad-

---

<sup>c</sup> Parallelization strategies will probably not pay off in the HF SCF framework, because, – as opposed to DFT – much of the value of a given HF SCF calculations comes from its nature as a reference for post-HF methods, which are unlikely to be feasible if already the HF SCF calculation cannot be carried out serially.

Note that a diagonalization-free, trust region-based HF and Kohn–Sham DFT “Augmented Root-haan–Hall” SCF algorithm has been proposed by Høst et al.<sup>155,166</sup> From its conceptual difference, however, an implementation in terms of the established QOL HF SCF modules is not straightforward, but surely a valuable endeavor.

<sup>d</sup> It cannot be finally judged here whether such implementation endeavors are independent of (future)

dressed at the 2-component level of theory because of spin component exchange coupling.

Very similar considerations apply to the 4-index integral transformation to the Fockian eigenbasis: The present implementation is limited by significant memory requirements and crosses the BLAS3 turnover point by far, and by far too fast with increasing system size. It is, therefore, logical to turn to integral-direct transformation techniques, i.e. to re-compute the integrals to be transformed on the fly. Any endeavors in this field will surely profit from progress made with the efficient re-implementation of the QOL's integral and iterator-evaluator modules.

As already stressed in sec. 4.1, the importance of this particular work is directly connected to the natural integration of the HF SCF (and, to some extent, the 4-index integral transformation) modules in the larger-scope Quantum Objects Library structure. Clearly, the generalization to the relativistic, 2-component pseudopotential framework, as done in this work mainly for the HF SCF modules, will have to be considered also, and in fact primarily, for the other, larger QOL parts.

---

considerations of molecular double group symmetry or not. In any case, such symmetry considerations would be both interesting and promising, particularly from the point of view of post-HF methods.

---



---

# Bibliography

1. Wang, S. G.; Qiu, Y. X.; Fang, H.; Schwarz, W. H. E.: *Chem. Eur. J.* **2006**, 12, 4101–4114.
2. Schwerdtfeger, P., Ed. *Relativistic Electronic Structure Theory. Part 1: Theory And Fundamentals*; Elsevier: Amsterdam, **2002**.
3. Schwerdtfeger, P., Ed. *Relativistic Electronic Structure Theory. Part 2: Applications*; Elsevier: Amsterdam, **2004**.
4. Thayer, J. S.: *J. Chem. Educ.* **2005**, 82, 1721–1727.
5. Bencini, A.: *Inorg. Chim. Acta* **2008**, 361, 3820–3831.
6. Marian, C. M.: In *Reviews In Computational Chemistry*; Lipkowitz, K. B., Boyd, D. B., Eds.; Wiley-VCH: New York, NY, **2001**; Vol. 15; pages 99–204.
7. Schwarz, W. H. E.: *J. Chem. Educ.* **2010**, 87, 444–448.
8. Wang, S.; Schwarz, W. H. E.: *Angew. Chem.* **2009**, 121, 3456–3467.
9. Neese, F.; Petrenko, T.; Ganyushin, D.; Olbrich, G.: *Coord. Chem. Rev.* **2007**, 251, 288–327.
10. Dyall, K. G.; Fægri, K.: *Introduction To Relativistic Quantum Chemistry*; Oxford University Press, Inc.: New York, NY, 1st ed., **2007**.
11. Reiher, M.; Wolf, A.: *Relativistic Quantum Chemistry*; Wiley-VCH: Weinheim, 1 ed., **2009**.
12. Hess, B. A.: *Phys. Rev. A* **1985**, 32, 756–763.
13. Hess, B. A.: *Phys. Rev. A* **1986**, 33, 3742–3748.

- 
14. Reiher, M.; Wolf, A.: *J. Chem. Phys.* **2004**, 121, 10945–10956.
  15. Reiher, M.; Wolf, A.: *J. Chem. Phys.* **2004**, 121, 2037–1047.
  16. Kutzelnigg, W.; Liu, W.: *J. Chem. Phys.* **2005**, 123, 241102-1–241102-4.
  17. Barysz, M.; Sadlej, A. J.; Snijders, J. G.: *Int. J. Quantum Chem.* **1997**, 65, 225–239.
  18. Barysz, M.; Sadlej, A. J.: *J. Chem. Phys.* **2002**, 116, 2696–2704.
  19. Dyall, K. G.: *J. Chem. Phys.* **1997**, 106, 9618–9626.
  20. van Lenthe, E.; Dyall, K. G.: *J. Chem. Phys.* **1999**, 111, 1366–1372.
  21. Quiney, H. M.; Skaane, H.; Grant, I. P.: *Adv. Quantum Chem.* **1999**, 32, 1–49.
  22. Lee, Y. S.; Ermler, W. C.; Pitzer, K. S.: *J. Chem. Phys.* **1977**, 67, 5861–5876.
  23. Hafner, P.; Schwarz, W. H. E.: *J. Phys. B* **1978**, 11, 217–233.
  24. Kahn, L. R.; Hay, P. J.; Cowan, R. D.: *J. Chem. Phys.* **1978**, 68, 2386–2398.
  25. Pitzer, R. M.; Winter, N. W.: *J. Phys. Chem.* **1988**, 92, 3061–3063.
  26. Ermler, W. C.; Lee, Y. S.; Christiansen, P. A.; Pitzer, R. M.: *Chem. Phys. Lett.* **1981**, 81, 70–74.
  27. Cao, X.; Dolg, M.: In *Challenges And Advances In Computational Chemistry And Physics: Relativistic Methods For Chemists*; Barysz, M., Ishikawa, Y., Eds.; Springer: Berlin; Heidelberg; New York, NY, **2010**; Vol. 10; pages 215–277.
  28. Hafner, P.; Schwarz, W. H. E.: *Chem. Phys. Lett.* **1979**, 65, 537–541.
  29. Hafner, P.: *J. Phys. B* **1980**, 13, 3297–3308.
  30. Lee, S. Y.; Lee, Y. S.: *J. Comput. Chem.* **1992**, 13, 595–601.
  31. McMurchie, L. E.; Davidson, E. R.: *J. Comput. Phys.* **1981**, 44, 289–301.
  32. Pitzer, R. M.; Winter, N. W.: *Int. J. Quantum Chem.* **1991**, 40, 773–780.
  33. Pitzer, R. M.: *J. Chem. Phys.* **1973**, 58, 3111–3112.
  34. Dupuis, M.; Rys, J.; King, H. F.: *J. Chem. Phys.* **1976**, 65, 111–116.
-

- 
35. Raffenetti, R. C.: *J. Chem. Phys.* **1973**, *58*, 4452–4458.
  36. Cancès, E.; Le Bris, C.: *Int. J. Quantum Chem.* **2000**, *79*, 82–90.
  37. Cancès, E.: In *Lecture Notes In Chemistry: Mathematical Models And Methods For Ab Initio Quantum Chemistry*; Defranceschi, M., Le Bris, C., Eds.; Springer: Berlin; Heidelberg; New York, NY, **2000**; Vol. 74; chapter 2, pages 17–43.
  38. Almlöf, J.; Faegri Jr., K.; Korsell, K.: *J. Comput. Chem.* **1982**, *3*, 385–399.
  39. Schrödinger, E.: *Phys. Rev.* **1926**, *28*, 1049–1970.
  40. Fließbach, T.: *Quantenmechanik*; Spektrum, Akad. Verlag: Heidelberg; Berlin, 3 ed., **2000**.
  41. Messiah, A.: *Quantenmechanik*, Vol. 1; Walter de Gruyter: Berlin; New York, 3 ed., **1990**.
  42. Schrödinger, E.: *Ann. Phys.* **1926**, *79*, 361–376.
  43. Schrödinger, E.: *Ann. Phys.* **1926**, *79*, 489–527.
  44. Schrödinger, E.: *Ann. Phys.* **1926**, *80*, 437–490.
  45. Schrödinger, E.: *Ann. Phys.* **1926**, *81*, 109–139.
  46. Domingos, J. M.: *J. Theor. Phys.* **1979**, *18*, 213–230.
  47. Messiah, A.: *Quantenmechanik*, Vol. 2; Walter de Gruyter: Berlin; New York, NY, 3 ed., **1990**.
  48. Wigner, E.: *Göttinger Nachrichten* **1932**, *31*, 546–559.
  49. Sachs, R. G.: *The Physics Of Time Reversal*; Chicago University Press: Chicago, **1987**.
  50. Lamb, J. S. W.; Roberts, J. A.: *Physica D* **1998**, *112*, 1–39.
  51. Lax, M.: *Symmetry Principles In Solid State And Molecular Physics*; Dover Publications: Mineola, NY, **2001**.
  52. Barron, L. D.; Buckingham, A. D.: *Acc. Chem. Res.* **2001**, *34*, 781–789.
  53. Pauli, W.: In *Niels Bohr And The Development Of Physics*; Pauli, W., Ed.; Pergamon Press: New York, NY, **1955**; pages 30–51.
  54. Kobayashi, M.: *Rev. Mod. Phys.* **2009**, *81*, 1019–1025.
-

- 
55. Maskawa, T.: Rev. Mod. Phys. **2009**, 81, 1027–1030.
  56. Swendsen, R. H.: J. Am. Phys. **2008**, 76, 643–648.
  57. Herbut, F.; Vujičić, M.: J. Math. Phys. **1967**, 8, 1345–1354.
  58. Wigner, E. P.: J. Math. Phys. **1960**, 1, 409–413.
  59. Wigner, E. P.: J. Math. Phys. **1960**, 1, 414–416.
  60. Kramers, H. A.: Proc. Acad. Amsterdam **1930**, 33, 959–972.
  61. Dirac, P. A. M.: Proc. Roy. Soc. London A **1928**, 117, 610–624.
  62. Schröder, U. E.: Spezielle Relativitätstheorie; Harri Deutsch Verlag: Frankfurt a.M., 4 ed., **2005**.
  63. Schwabl, F.: Quantenmechanik Für Fortgeschrittene; Springer: Berlin; Heidelberg; New York, NY, 3 ed., **2004**.
  64. Dirac, P. A. M.: Proc. Roy. Soc. London A **1929**, 126, 360–365.
  65. Dirac, P. A. M.: Proc. Roy. Soc. London A **1931**, 131, 610–624.
  66. Norrby, L. J.: J. Chem. Educ. **1991**, 86, 110–113.
  67. Szabo, A.: J. Chem. Educ. **1969**, 46, 678.
  68. Chang, C.; Pelissier, M.; Durand, P.: Phys. Scr. **1986**, 34, 394–404.
  69. van Lenthe, E.; Baerends, E. J.; Snijders, J. G.: J. Chem. Phys. **1993**, 99, 4597–4610.
  70. van Lenthe, E.; Baerends, E. J.; Snijders, J. G.: J. Chem. Phys. **1994**, 101, 9783–9792.
  71. Foldy, L. L.; Wouthuysen, S. A.: Phys. Rev. **1950**, 78, 29–36.
  72. Douglas, M.; Kroll, N. M.: Ann. Phys. **1974**, 82, 89–155.
  73. Cowan, R. D.; Griffin, D.: J. Opt. Soc. Am. **1976**, 66, 1010–1014.
  74. Wood, J. H.; Boring, A. M.: Phys. Rev. B **1978**, 18, 2701–2711.
  75. Dolg, M.: In Relativistic Electronic Structure Theory. Part 1: Fundamentals; Theoretical and Computational Chemistry; Schwerdtfeger, P., Ed.; Elsevier: Amsterdam, **2002**; Vol. 11; chapter 14, pages 793–862.
  76. Breit, G.: Phys. Rev. **1929**, 34, 553–573.
  77. Breit, G.: Phys. Rev. **1932**, 39, 616–624.
-

- 
78. Kutzelnigg, W.: *Phys. Scripta* **1987**, 36, 416–431.
  79. Dyall, K. G.: *J. Chem. Inf. Comput. Sci.* **2001**, 41, 30–37.
  80. Stoll, H.; Metz, B.; Dolg, M.: *J. Comput. Chem.* **2002**, 130, 164108-1–164108-12.
  81. Hellmann, H.: *J. Chem. Phys.* **1935**, 3, 61.
  82. Gombás, P.: *Z. Phys.* **1935**, 94, 473–488.
  83. Phillips, J. C.; Kleinman, L.: *Phys. Rev.* **1959**, 116, 287–294.
  84. Weeks, J. D.; Rice, S. A.: *J. Chem. Phys.* **1968**, 49, 2741–2756.
  85. Berthier, G.; Defranceschi, M.; Le Bris, C.: *Int. J. Quantum Chem.* **2003**, 93, 156–165.
  86. Kahn, L. R.; Goddard, W. A.: *J. Chem. Phys.* **1972**, 56, 2685–2701.
  87. Kahn, L. R.; Baybutt, P.; Truhlar, D. G.: *J. Chem. Phys.* **1976**, 65, 3826–3853.
  88. Hartree, D. R.: *Proc. Cambr. Phil. Soc.* **1928**, 24, 89–110.
  89. Hartree, D. R.: *Proc. Cambr. Phil. Soc.* **1928**, 24, 111–132.
  90. Hartree, D. R.: *Proc. Cambr. Phil. Soc.* **1928**, 24, 426–437.
  91. Fock, V.: *Z. Phys.* **1930**, 61, 126–148.
  92. Slater, J. C.: *Phys. Rev.* **1930**, 35, 210–.
  93. Primas, H.; Müller-Herold, U.: *Elementare Quantenchemie*; Teubner Verlag: Stuttgart, **1990**.
  94. Ostrovsky, V. N.: *HYLE – Int. J. Phil. Chem.* **2005**, 11, 101–126.
  95. Schwarz, W. H. E.: *Angew. Chem.* **2006**, 118, 1538–1547.
  96. Scerri, E. R.: *J. Chem. Educ.* **2000**, 77, 522–525.
  97. Scerri, E. R.: *Found. Chem.* **1999**, 1, 297–305.
  98. Schwarz, W. H. E.; Rich, R. L.: *J. Chem. Educ.* **2010**, 87, 435–443.
  99. Schwarz, W. H. E.: *Theor. Chem. Acc.* **2001**, 105, 271–275.
  100. Szabo, A.; Ostlund, N. S.: *Modern Quantum Chemistry: Introduction To Advanced Electronic Electronic Structure Theory*; Dover Publications: Mineola, NY, **1996**.
-

- 
101. Helgaker, T.; Jørgensen, P.; Olsen, J.: Molecular Electronic Structure Theory; John Wiley & Sons, Ltd.: Chichester, **2000**.
  102. Kohn, W.; Sham, L. J.: Phys. Rev. A **1965**, 140, 1133–1138.
  103. Koch, W.; Holthausen, M. C.: A Chemist’s Guide To Density Functional Theory; Wiley-VCH: Weinheim, 2 ed., **2001**.
  104. Parr, R. G.; Yang, W.: Density-Functional Theory Of Atoms And Molecules, Vol. 16 of International Series Of Monographs On Chemistry; Oxford University Press, Inc.: New York, NY, **1989**.
  105. Almlöf, J.: In Lecture Notes In Chemistry; Roos, B. O., Ed.; Springer: Berlin; Heidelberg; New York, NY, **1994**; Vol. 64; chapter 1, pages 1–90.
  106. Defranceschi, M.; Le Bris, C.: J. Math. Chem. **1997**, 21, 1–30.
  107. Echenique, P.; Alonso, J. L.: Mol. Phys. **2007**, 105, 3057–3098.
  108. Lieb, E. H.; Simon, B.: Commun. Math. Phys. **1977**, 53, 185–194.
  109. Lions, P. L.: Comm. Math. Phys. **1987**, 109, 33–97.
  110. Lieb, E. H.; Simon, B.: J. Chem. Phys. **1974**, 61, 735–736.
  111. Rösch, N.: Chem. Phys. **1983**, 80, 1–5.
  112. Roothaan, C. C. J.: Rev. Mod. Phys. **1951**, 22, 69–89.
  113. Hall, G. G.: Proc. Roy. Soc. London A **1951**, 205, 541–552.
  114. Kobus, J.: Adv. Quantum Chem. **1997**, 28, 1–14.
  115. Boys, S. F.: Proc. Roy. Soc. (London) A **1950**, 200, 542–554.
  116. Armbruster, M. K.; Klopper, W.; Weigend, F.: Phys. Chem. Chem. Phys. **2006**, 8, 4862–4865.
  117. Helgaker, T.; Klopper, W.; Tew, D. P.: Mol. Phys. **2008**, 106, 2107–2143.
  118. Rayleigh, J. W. S.: Theory Of Sound, Vol. 1; McMillan: London, 2 ed., **1894**.
  119. Møller, C.; Plesset, M. S.: Phys. Rev. **1934**, 46, 618–622.
  120. Cary, J. R.; Shasharina, S. G.; Cummings, J. C.; Reynders, J. V. W.; Hinkler, P. J.: Comp. Phys. Comm. **1997**, 105, 20–36.
  121. Häser, M.; Ahlrichs, R.: J. Comp. Chem. **1989**, 10, 104–111.
-

- 
122. Obara, S.; Saika, A.: J. Chem. Phys. **1986**, 84, 3963–3974.
123. Obara, S.; Saika, A.: J. Chem. Phys. **1988**, 89, 1540–1559.
124. Culbert, C.: J. Gen. Lie Theory Appl. **2007**, 1, 1–17.
125. Daboul, J.; Delbourgo, R.: J. Math. Phys. **1999**, 40, 4134–4150.
126. Girard, P. R.: Quaternions, Clifford Algebras And Relativistic Physics; Birkhäuser Verlag AG: Basel; Bonn; Berlin, 1st ed., **2007**.
127. Saue, T.; Jensen, H. J. A.: J. Chem. Phys. **1999**, 111, 6211–6222.
128. Jensen, H. J. A.; Dyall, K. G.; Saue, T.; Fægri, K.: J. Chem. Phys. **1996**, 104, 4083–4097.
129. Fischer, G.: Lineare Algebra; Vieweg & Sohn Verlag: Wiesbaden, 15 ed., **2005**.
130. Peterson, K. A.; Figgen, D.; Goll, E.; Stoll, H.; Dolg, M.: J. Chem. Phys. **2003**, 119, 11113–11123.
131. Metz, B.; Schweizer, M.; Stoll, H.; Dolg, M.; Liu, W.: Theor. Chem. Acc. **2000**, 104, 22–28.
132. Lischka, H.; Shepard, R.; Brown, F. B.; Shavitt, I.: Int. J. Quantum Chem., Quantum Chem. Symp. **1981**, 20, 91–100.
133. Shepard, R.; Shavitt, I.; Pitzer, R. M.; Comeau, D. C.; Pepper, M.; Lischka, H.; Szalay, P. G.; Ahlrichs, R.; Brown, F. B.; Zhao, J.: Int. J. Quantum Chem. **1988**, 22, 149–165.
134. Lischka, H.; Shepard, R.; Pitzer, R. M.; Shavitt, I.; Dallos, M.; Müller, T.; Szalay, P. G.; Seth, M.; Kedziora, G. S.; Yabushita, S.; Zhang, Z.: Phys. Chem. Chem. Phys. **2001**, 3, 664–673.
135. Lischka, H.; Shepard, R.; I.Shavitt.; Pitzer, R. M.; Dallos, M.; Müller, T.; Szalay, P. G.; Brown, F. B.; Ahlrichs, R.; Böhm, H. J.; Chang, A.; Comeau, D. C.; Gdanitz, R.; Dachsel, H.; Erhardt, C.; Ernzerhof, M.; Höchtl, P.; Irle, S.; Kedziora, G.; Kovar, T.; Parasuk, V.; Pepper, M. J. M.; Scharf, P.; Schiffer, H.; Schindler, M.; Schüler, M.; Seth, M.; Stahlberg, E. A.; Zhao, J.; Yabushita, S.; Zhang, Z.; Barbatti, M.; Matsika, S.; Schüürmann, M.; Yarkony, D. R.; Brozell, S. R.; Beck, E. V.; Blaudeau, J.: Columbus, an *ab initio* electronic structure program, release 5.9.1 **2006**.
-

- 
136. Ahlrichs, R.: *Phys. Chem. Chem. Phys.* **2004**, 6, 5119–5121.
137. Sierka, M.; Hogekamp, A.; Ahlrichs, R.: *J. Chem. Phys.* **2003**, 118, 9136–.
138. Huber, K. P.; Herzberg, G.: *Molecular Spectra And Molecular Structure*, Vol. IV. Constants Of Diatomic Molecules; Van Nostrand Reinhold Co.: New York, NY, **1979**.
139. Peterson, K. A.: *J. Chem. Phys.* **2003**, 119, 11099–11112.
140. Dunning Jr., T. H.: *J. Chem. Phys.* **1989**, 90, 1007–1023.
141. Park, B. H.; Han, Y. K.; Park, S. J.; Lee, Y. S.: *Int. J. Quantum Chem.* **1998**, 66, 91–98.
142. Abdulnur, S. F.; Linderberg, J.; Öhrn, U.; Thulstrup, P. W.: *Phys. Rev. A* **1972**, 6, 889–898.
143. Jørgensen, P.; Öhrn, Y.: *Phys. Rev. A* **1973**, 8, 112–119.
144. Cancès, E.; Le Bris, C.: *ESAIM: Mathematical Modelling And Numerical Analysis* **2000**, 34, 749–774.
145. Schwarz, W. H. E.; Wang, S.: *Int. J. Quantum Chem.* **2010**, 110, 1455–1465.
146. Figgen, D.; Peterson, K. A.; Dolg, M.; Stoll, H.: *J. Chem. Phys.* **2009**, 130, 164108-1–164108-12.
147. Hartree, D. R.: *The Calculation Of Atomic Structures*; John Wiley & Sons, Ltd.: Chichester, **1957**.
148. Karlström, G.: *Chem. Phys. Lett.* **1979**, 67, 348–350.
149. Zerner, M.; Hehenberger, M.: *Chem. Phys. Lett.* **1979**, 62, 550–554.
150. Pulay, P.: *J. Comp. Chem.* **1982**, 3, 556–560.
151. Kudin, K. N.; Scuseria, G. E.; Cancès, E.: *J. Chem. Phys.* **2002**, 116, 8255–8261.
152. Kudin, K. N.; Scuseria, G. E.: *ESAIM: Mathematical Modelling And Numerical Analysis* **2007**, 42, 281–296.
153. Amat, L.; Carbó-Dorca, R.: *Int. J. Quantum Chem.* **2002**, 87, 59–67.
154. Szekeres, Z.; Mezey, P. G.; Surján, P. R.: *Chem. Phys. Lett.* **2006**, 424, 420–424.
-

- 
155. Jansík, B.; Høst, S.; Johannson, M. P.; Olsen, J.; Jørgensen, P.; Helgaker, T.: *J. Chem. Theory Comput.* **2009**, 5, 1027–1032.
  156. van Lenthe, J. H.; Zwaans, R.; van Dam, H. J. J.; Guest, M. F.: *J. Comput. Chem.* **2006**, 27, 926–932.
  157. Tang, K. C.; Edmiston, C.: *J. Chem. Phys.* **1970**, 52, 997–998.
  158. Saunders, V. R.; van Lenthe, J. H.: *Mol. Phys.* **1983**, 48, 923–954.
  159. Hurley, J. N.; Huestis, D. L.; Goddard III, W. A.: *J. Phys. Chem.* **1988**, 92, 4880–4883.
  160. Yamamoto, S.; Nagashima, U.: *Comp. Phys. Comm.* **2005**, 116, 58–65.
  161. Runeberg, N.; Pyykkö, P.: *Int. J. Quantum Chem.* **1998**, 66, 131–140.
  162. de Laeter, J. R.; Böhlke, J. K.; De Bièvre, P.; Hidaka, H.; Preiser, H. S.; Rosman, K. J. R.; Talyor, P. D. P.: *Pure Appl. Chem.* **2003**, 75, 683–800.
  163. Storchi, L.; Belpassi, L.; Tarantelli, F.; Sgamellotti, A.; Quiney, H. M.: *J. Chem. Theory Comput.* **2010**, 6, 384–394.
  164. Armbruster, M. K.; Weigend, F.; van Wüllen, C.; Klopper, W.: *Phys. Chem. Chem. Phys.* **2008**, 10, 1478–1756.
  165. Armbruster, M. K.; Schimmelpfennig, B.; Plaschke, M.; Rothe, J.; Denecke, M. A.; Klenze, R.: *J. Electron Spectrosc. Relat. Phenom.* **2008**, 169, 51–56.
  166. Høst, S.; Olsen, J.; Jansík, B.; Thøgersen, L.; Jørgensen, P.; Helgaker, T.: *J. Chem. Phys.* **2008**, 129, 124106-1–124106-12.

---

# List of Figures

|            |                                                                                                                     |     |
|------------|---------------------------------------------------------------------------------------------------------------------|-----|
| <b>4.1</b> | UML class diagrams for parts of the QOL::MatrixVector and QOL::LinearAlgebra namespaces                             | 44  |
| <b>4.2</b> | UML class diagrams for parts of the QOL::LinearAlgebra and QOL::CartesianGaussianIntegration namespaces             | 49  |
| <b>5.1</b> | UML class diagrams for the new HermitianCayleyMatrix and HermitianTimeReversalInvariantMatrixRepresentation classes | 57  |
| <b>5.2</b> | Layout of the HermitianCayleyMatrix_MemEvaluator's STL vector-type container class                                  | 58  |
| <b>5.3</b> | Construction of spin-free operator matrix representations                                                           | 67  |
| <b>5.4</b> | Matrix visualization: 2-component HF SCF density matrices                                                           | 68  |
| <b>6.1</b> | QOL pseudopotential parameter definition for the $g_1\text{Ti}$ small-core MC-DHF PP of Metz et al. <sup>131</sup>  | 74  |
| <b>6.2</b> | Construction of general 2-component pseudopotential operator matrix representations                                 | 89  |
| <b>7.1</b> | Mean assemble_2cSCF2ParticleParts CPU times $t$                                                                     | 94  |
| <b>7.2</b> | Angular symmetry breakdown and removeSymmetryContamination                                                          | 105 |
| <b>7.3</b> | Pictorial representations of eigenvector and density matrices                                                       | 106 |
| <b>7.4</b> | Performance of the SCF ODA for the $72\text{Hf}$ atom                                                               | 110 |
| <b>7.5</b> | Comparison of Roothaan–Hall-type and SCF ODA convergence behavior                                                   | 111 |
| <b>7.6</b> | Comparison of core Hamiltonian and MFAD initial guess SCF ODA convergence                                           | 115 |
| <b>8.1</b> | Illustration of a single pair transformation step                                                                   | 121 |

---

|            |                                                                                                                  |     |
|------------|------------------------------------------------------------------------------------------------------------------|-----|
| <b>8.2</b> | Illustration of the 2-component 4-index integral transformation algorithm                                        | 123 |
| <b>8.3</b> | Difference $\Delta E$ of the $\text{Rn}_2$ total MP2 energy $E_{\text{MP2}}$ and the respective monomer energies | 130 |

## List of Tables

|            |                                                                                                                                                                               |     |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>6.1</b> | Ordering of Cartesian monomials $\mathbf{m} = (m_x, m_y, m_z)$ by ARGOS                                                                                                       | 82  |
| <b>6.2</b> | Variable types, names, and meanings of the EDIT subroutine's parameter list.                                                                                                  | 85  |
| <b>7.1</b> | Construction of $\mathbf{K} \in \mathbb{C}^{2n \times 2n}$ by <code>assemble_2cSCF2Particle-Parts</code>                                                                      | 92  |
| <b>7.2</b> | Construction of the auxiliary matrices $\mathbf{k}_{\alpha\alpha}, \mathbf{k}_{\alpha\beta} \in \mathbb{C}^{n \times n}$                                                      | 93  |
| <b>7.3</b> | Numbers of SCF iterations $v$ needed to converge $E_{\text{SCF}}$ below $\delta_E = 10^{-9}$ a.u. for the TIH case                                                            | 112 |
| <b>7.4</b> | Comparison of core Hamiltonian and MFAD initial guess for the TIH example of fig. 7.6                                                                                         | 115 |
| <b>8.1</b> | Re-writing the Kramers-restricted MP2 energy in terms of the representative integrals computed within the 2-component 4-index integral transformation algorithm of sec. 8.2.2 | 129 |
| <b>8.2</b> | Equilibrium distances $d_e$ , dissociation energies $D_e$ , and harmonic frequencies $\omega_e$ for the $^{222}_{86}\text{Rn}$ dimer                                          | 130 |

---

# Acronyms and Abbreviations

The following acronyms and abbreviations are used throughout this work. Whereas most are conventional and common in the context of relativistic quantum chemistry and computer science, some are not, such that a complete list is given.<sup>e</sup>

---

|            |                                                       |
|------------|-------------------------------------------------------|
| AS         | Atomic 2-Spinor                                       |
| BSSE       | Basis Set Superposition Error                         |
| CC         | Coupled Cluster                                       |
| CCSD       | Coupled Cluster with Single and Double excitations    |
| CCSD(T)    | CCSD with non-iterative Triple excitation corrections |
| CGF        | Cartesian Gaussian Function                           |
| CGTO       | Cartesian Gaussian-Type Orbital                       |
| CI         | Configuration Interaction                             |
| <i>CPT</i> | “Charge–Parity–Time”                                  |
| DC         | Dirac–Coulomb                                         |
| DCB        | Dirac–Coulomb–Breit                                   |
| DHF        | Dirac–Hartree–Fock                                    |
| DIIS       | Direct Inversion of the Iterative Subspace            |
| DFT        | Density Functional Theory                             |
| EDIIS      | Energy-Direct Inversion of the Iterative Subspace     |
| GPK        | Generalized Philips–Kleinman                          |
| GCHF       | Grand Canonical Hartree–Fock                          |

---

<sup>e</sup> The chosen capitalization emphasizes the meaning of the respective acronym or abbreviation and is, thus, neither systematic nor meant to be orthographically correct.

---

|         |                                                      |
|---------|------------------------------------------------------|
| GCKS    | Grand Canonical Kohn–Sham                            |
| GTO     | Gaussian-Type Orbital                                |
| HF      | Hartree–Fock                                         |
| LARL(1) | Lookahead-LR (parser with lookahead 1)               |
| MC-     | Multi-Configuration-                                 |
| MFAD    | Molecule-From-Atoms Density                          |
| MP2     | Møller–Plesset Perturbation Theory to Second Order   |
| MS      | Molecular 2-Spinor                                   |
| ODA     | Optimal Damping Algorithm                            |
| PP      | Pseudopotential                                      |
| QED     | Quantum Electrodynamics                              |
| QOL     | “Quantum Objects Library”                            |
| SCF     | Self-Consistent Field                                |
| STL     | Standard Template Library                            |
| SO-CIDS | Spin–Orbit Configuration Interaction Singles Doubles |
| UML     | Unified Modeling Language                            |
| XML     | Extended Markup Language                             |

---

---

## **Erklärung gemäß §4 Abs. 1, 9. der Promotionsordnung**

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde.

Die Bestimmungen dieser Promotionsordnung sind mir bekannt.

Die von mir vorgelegte Dissertation ist von Prof. Dr. M. Dolg betreut worden.

---

Datum

---

Unterschrift

---

# Curriculum Vitae

## Persönliche Angaben:

Dipl.-Chem. Jonas Wiebke  
**geboren:** 22. März 1980, Frankfurt a. M.  
**Familienstand:** ledig

## Beruflicher bzw. Wissenschaftlicher Werdegang

- 1986–1990** G. Weerth-Grundschule, Detmold  
**1990–1999** Chr.-D. Grabbe-Gymnasium, Detmold  
**08. 1999–06. 2000** Ersatzdienst Klinikum Lippe-Detmold GmbH,  
stationäre Krankenpflege  
**10. 2000–06. 2006** Studium Chemie, Universität Köln  
■ **12. 2003:** Vordiplom  
■ **06. 2006:** Abschluss (Dipl.-Chem., Auszeichnung).  
Diplomarbeit: “*Hydratation der dreiwertigen Ionen der Actinoide*”, Betreuer: M. Dolg  
**07. 2006–07. 2010** Promotionsstudium Theoretische Chemie, Universität  
zu Köln.  
Dissertation: “*Kramers-Restricted Self-Consistent  
2-Spinor Fields for Heavy Element Chemistry*”,  
Betreuer: M. Dolg

---

## Acknowledgments

- First of all, I wish to thank M. Dolg for his continuous, generous, and faithful support in a large number of scientific, technical ,and personal questions. Over the last years, he proved to be a great teacher and a caring supervisor for me. It has always been a pleasure to work for him, and in his group!
- I also thank M. Hanrath for his professional and patient help with technical and theoretical issues, and his many lessons on C++ and computer science. He had to repeat a lot of things more often than twice, but, as he knows, “*es ist nicht jedem gegeben.*”
- U. Deiters is acknowledged, because he agreed to assess this work, and because he helped and supported me a lot in during the application for post-doctoral grants. Thank you!
- R. M. Pitzer of (formerly) Ohio State University is greatly acknowledged for providing the ARGOS integral program source code and his kind permission to use it.
- In particular, I wish to thank both my office mates Anna Engels-Putzka and Joachim Friedrich for their company, understanding, support, the numerous discussions on mathematics, physics, and chemistry, and the fun. Both read parts of the manuscript of this work and made important comments.
- All present and former members of Cologne’s theoretical chemistry group, i.e.: M. Böhler, B. Börsch-Pulm, M. Burkatzki, X. Cao-Dolg, J. Ciupka, R. Fondermann, T. Hangele, D. Heidelberg, J. Held, M. Hülsen, S. Ketrat, D. Pape, A. Schnurpeil, K. Walczak, A. Weigand, D. Weissmann, and J. Yang, for the great company and the good atmosphere at the institute.
- I am very grateful to a very large number of people who shared my time at Cologne’s Chemistry Department and who made me feel at home there sometimes. It is impossible to name all of them. I therefore wish to thank only some of them explicitly, i.e.: Kristina Hackelöer, Andrea Hunold, Patrick Jacobs, Slawomir Pitula, and Davor Turkovic, for a great time; and particularly Jutta Schütte for being a closest friend for 10 years.

- 
- I wish to express my deep gratitude to my parents Gudrun and Claus.  
Nothing of what is important to me today would have been possible if they were not exactly the people they are.
  - Finally, I wish to thank Birthe, who is the most wonderful person I know.  
Without her faith, her strength, her love, and her way to be I would surely be somewhere else, somebody else, less happy with life.