
Building Effective Models for Correlated Electron Systems

Von der Fakultät für Mathematik, Informatik und
Naturwissenschaften der RWTH Aachen University zur Erlangung des
akademischen Grades eines Doktors der Naturwissenschaften
genehmigte Dissertation

vorgelegt von
Qian Zhang, M.Sc.
aus
Guiyang, Guizhou, China

Berichter: Prof. Dr. Erik Koch
Prof. Dr. Carsten Honerkamp

Tag der mündlichen Prüfung: 17. 08. 2020

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek
online verfügbar.

Abstract

To understand strongly correlated systems, we must confront the many-body problem. This is practically impossible for the ab-initio Hamiltonian. To make such studies feasible it is, thus, crucial to construct model Hamiltonians that are as simple as possible, so they can be solved, while containing still enough details to be material-specific.

Our starting point is density functional theory for individual atoms and ions to obtain realistic basis functions and the corresponding matrix elements. For the open-shell orbitals, which show the strongest correlation effects due to the degeneracy of the multiplets, we calculate the Slater-Condon and spin-orbit parameters from the resulting self-consistent radial wave functions and potentials. We study the trends of the parameters systematically across the periodic table, develop an intuitive parametrization, and calculate atomic open-shell spectra in LS -, intermediate-, and jj -coupling schemes.

The comparison of the interaction strengths of different coupling schemes gives rise to the study of the moment formulas, which reduce the calculation from the “impossible” many-electron Hilbert space to a one- or two-electron space. We derive the analytic moment formulas for the general one- and two-body Hamiltonians. The moment formulas provide us a new approach to handle the many-electron Hamiltonians without the need of working with a many-electron basis, but only with matrix representations under the one- or two-electron basis.

To model the Hamiltonians for realistic materials, orthonormal basis orbitals are preferred. However, while the atomic orbitals are mutually orthonormal within a single atom, they are, in general, non-orthogonal for atoms on different lattice sites. We study and develop efficient multi-center integral techniques for evaluating orbital overlaps, which are essential for performing the orbital orthogonalization. To orthogonalize the basis orbitals, we apply the Löwdin symmetric orthogonalization scheme, which minimizes the orbital modification. To generalize the multi-center integrals, we introduce the re-centering method, which is a spherical harmonic expansion that requires the Gaunt coefficients with large angular quantum numbers. To compute the Gaunt coefficients, the previously known numerical methods are, however, inaccurate for the coefficients that involve large quantum numbers. Therefore, we provide a numerically stable algorithm for computing the Gaunt coefficients efficiently and accurately. The re-centering method enables us to compute general multi-center integrals including the hopping matrix elements and the long-range Coulomb matrix elements. After performing the basis orthonormalization, we study the deformation of the resulting orbitals and investigate the modification of the corresponding multi-center matrix elements under changes of the bond lengths or lattice constants.

Zusammenfassung

Wollen wir stark korrelierte Materialien verstehen, sehen wir uns mit dem Vielteilchen-Problem konfrontiert. Für parameterfreie ab-initio Beschreibungen ist dieses Problem in der Praxis unlösbar. Um dennoch stark korrelierte Materialien simulieren zu können ist es daher essentiell Modelle zu entwickeln, die einfach genug sind, um mit vertretbarem Aufwand lösbar zu sein, die aber immer noch komplex genug sind, um materialspezifisch zu sein.

Wir beginnen die Konstruktion realistischer Modelle mit Dichtefunktional-Rechnungen für einzelne Atome, mittels derer wir realistische Basisfunktionen sowie die zugehörigen Matrix-Elemente bestimmen. Für offenschalige Systeme, die aufgrund ihrer hohen Entartung die stärksten Korrelations-Effekte zeigen, berechnen wir die Slater-Condon und Spin-Orbit Parameter aus den selbstkonsistenten Wellenfunktionen und Potenzialen. Wir untersuchen die Trends dieser Parameter über das gesamte Periodensystem, entwickeln eine intuitive Parametrisierung und bestimmen die atomaren Spektren in LS -, gemischter und in jj -Kopplung.

Um die Stärke der Wechselwirkungen sinnvoll vergleichen zu können, untersuchen wir die spektrale Aufspaltung mittels ihrer Varianz. Dazu entwickeln wir allgemeine analytische Ausdrücke für die Momente des Hamilton-Operators: Formeln für beliebige Momente des Einteilchen- sowie für Momente beliebiger Zweiteilchen-Operatoren bis zur Ordnung zwei. Diese bemerkenswerten Formeln erlauben es uns die Breite der Vielteilchen-Spektren extrem effizient und völlig ohne Betrachtung des Vielteilchen-Hilbertraums zu bestimmen. Dabei arbeiten wir mit Matrizen, deren Dimension lediglich mit dem Quadrat der Anzahl der Basisfunktionen wächst.

Die Modell-Beschreibung realistischer Materialien wird intuitiv und vermeidet große Überlapp-Matrizen wenn die Basisfunktionen orthonormal sind. Atomare Funktionen sind zwar orthogonal auf ihrem jeweiligen Platz, haben aber einen nicht-verschwindenden Überlapp zwischen verschiedenen Gitterplätzen. Wir entwickeln effiziente Multizentren-Techniken zur Berechnung dieser Überlapp-Integrale, die die Voraussetzung für die Orthogonalisierung bilden. Wir nutzen das Löwdinsche Orthogonalisierungs-Verfahren, das eine minimale Modifikation der Orbitale garantiert. Wir verallgemeinern die Multizentren-Integrale durch Einführung der Rezentrierungs-Methode bei der die Orbitale um einen anderen Gitterplatz in Kugelflächenfunktionen entwickelt werden. Für verlässliche Rechnungen benötigen wir dazu Gaunt Koeffizienten für sehr hohe Drehimpuls-Quantenzahlen. Da die bekannten Methoden dazu nicht genau genug sind, entwickeln wir einen numerisch stabilen Algorithmus, der genauer und gleichzeitig effizienter als andere Methoden ist. Die Rezentrierungs-Methode erlaubt es uns schließlich, die allgemeinen Multizentren-Integrale für die Hüpf-Integrale und die langreichweitigen Coulomb-Integrale zu bestimmen. Als Beispiel für den Effekt der Basis-Orthogonalisierung untersuchen wir die Deformation der atomaren Orbitale und die daraus resultierende Änderung der Modell-Parameter bei Änderung der Bindungslängen bzw. Gitterkonstanten.

Overview of the thesis

Chapter 1: What is a many-electron problem? In this chapter, we introduce the one-electron orbital basis, on which the second quantization formalism is built. We discuss the construction of a many-electron basis and the set up of the matrix representations of the Hamiltonians. The keywords are: *second quantization, bit representation, many-electron basis, matrix representations*.

Chapter 2: How to use a one-electron orbital basis to describe solids? In this chapter, we discuss the tight-binding method and the general idea of using the method to describe realistic materials. The keywords are: *tight-binding, Bloch waves, energy bands*.

Chapter 3: How to obtain realistic atomic orbitals and potentials? In this chapter, we solve many-electron atomic systems in the framework of density functional theory (DFT). The keywords are: *atomic orbitals, Kohn-Sham equation, self-consistent fields, local density approximation*.

Chapter 4: How to use the *LS*- and *jj*-coupling schemes to solve many-electron problems? In this chapter, we solve atomic open-shell many-electron problems analytically, using different coupling schemes. The keywords are: *open-shell, atomic multiplets, LS-coupling, jj-coupling, Coulomb Hamiltonian, spin-orbit Hamiltonian*.

Chapter 5: Is there a pattern among the matrix elements? In this chapter, we study systematically the trends of the Coulomb and spin-orbit interactions and the corresponding matrix elements in the atomic open-shell systems over the entire periodic table. The keywords are: *trends, Slater-Condon parameters, spin-orbit parameters, spectral variance*.

Chapter 6: Is it possible to extract physical information from a many-electron Hamiltonian without working with a many-electron basis? In this chapter, we present an analytic approach to compute the moments (e.g. mean and variance) of a general many-electron Hamiltonian. The keywords are: *moments, one- and two-body Hamiltonians*.

Chapter 7: How to compute the Gaunt coefficients? When calculating Coulomb matrix elements or multi-center integrals we must evaluate Gaunt coefficients. In this chapter, we develop a systematic approach based on finite-precision arithmetic to compute the Gaunt coefficients efficiently and accurately, even for large angular momenta. The keywords are: *algorithm for Gaunt coefficients, spherical harmonics*.

Chapter 8: What if the basis orbitals are not orthogonal? In this chapter, we discuss a systematic way to compute orbital overlaps. From the overlap matrix, we can produce a new set of basis orbitals which are mutually orthonormal, yet preserving their atomic features as much as possible. The keywords are: *orbital overlap, Fourier transformation*,

Löwdin symmetric orthogonalization.

Chapter 9: How to compute matrix elements that involve multiple centers? In this chapter, we discuss the evaluation of multi-center integrals including hopping matrix elements and long-range Coulomb matrix elements. We study the effects of orbital orthogonalization on the resulting matrix elements. The keywords are: *multi-center integrals, re-centering method, hopping matrix elements, long-range Coulomb matrix elements.*

Python codes:

In the thesis, I put effort on explaining how things are implemented in practice. For important methods and algorithms, I present my solutions in Python (version 3), which is chosen for its code readability. Sometimes I receive comments like, “*Qian, no one will read your code!*” I agree on that I am not presenting the “best solutions”. However, I believe there is always a group of audiences who want to see how things are actually implemented. The process of translating a physical problem into precise machine instructions is in some sense breaking down a complicated problem into simple tasks, which often helps us better understand a problem, realize where the crucial points or the difficulties are, and potentially helps us gain new insights to the problems.

Contents

1. Many-electron states	1
1.1. Introducing a basis	1
1.2. Representation of a many-electron basis	4
1.3. Representation of one- and two-body Hamiltonians	9
2. The tight-binding method	15
2.1. The tight-binding Hamiltonian	15
2.2. Bloch's theorem	17
2.3. One-dimensional lattice problem	18
2.4. Multi-band one-dimensional lattice problem	21
2.5. The electronic structure of graphene	23
2.6. Carbon nanotube	26
3. Self-consistent field solver for atomic systems	31
3.1. The starting point	31
3.2. Hydrogen-like system	31
3.3. Logarithmic grid	33
3.4. Numerov's method	35
3.5. The shooting and matching methods	35
3.6. Predict ΔE by first-order perturbation theory	38
3.7. The eigen-state solver	43
3.8. Many-electron atomic system and the Kohn-Sham equation	47
3.9. Hartree and exchange-correlation potentials	49
3.10. Total energy of the many-body system	51
3.11. The self-consistent field solver	53
4. LS-coupling and jj-coupling schemes for atomic open-shell problems	59
4.1. Atomic open-shell problems	59
4.2. LS -basis and jj -basis	60
4.3. The Coulomb and spin-orbit matrix elements	63
4.4. The LS -coupling scheme	65
4.4.1. Diagonalizing H_U in the LS -basis	65
4.4.2. Diagonalizing H_{SO} in the LS -basis	75
4.5. The jj -coupling scheme	78
4.5.1. Diagonalizing H_{SO} in the jj -basis	78
4.5.2. Diagonalizing H_U in the jj -basis	79
4.6. Transformation between LS -basis and jj -basis states	84

4.7. The intermediate-coupling scheme	87
5. Trends of many-body effects in atomic open-shells	91
5.1. Trends of the matrix elements	91
5.2. DFT calculations and the structure of the periodic table	92
5.3. Trends of Slater-Condon parameters	94
5.4. The Hubbard U_{avg} and Hund's exchange J_{avg}	103
5.5. Trends of spin-orbit parameters	106
5.6. Trends of the Coulomb and spin-orbit interaction	109
6. Moments of many-electron Hamiltonians	113
6.1. Spectral moments	113
6.2. Moments of one-body Hamiltonians	114
6.2.1. The first moment	114
6.2.2. The second moment	115
6.2.3. The third moment	117
6.2.4. The n -th moment	121
6.3. Moments of two-body Hamiltonians	123
6.3.1. The first moment	123
6.3.2. The second moment	124
7. Recursive computation of Gaunt coefficients	133
7.1. Gaunt coefficients	133
7.2. Symmetry properties	133
7.3. Explicit formula	134
7.4. Gaunt coefficients arranged as a matrix	137
7.5. Recursion relations	138
7.6. Re-ordering	142
7.7. Error comparison among different algorithms	146
7.8. List of the first few Gaunt coefficients in matrix form	148
8. Orbital overlaps and basis orthonormalization	151
8.1. Orbital overlaps	151
8.2. Plane waves and spherical waves	152
8.2.1. Expand a plane wave into spherical waves	153
8.2.2. Expand a spherical wave into plane waves	155
8.2.3. Expand an atomic orbital into spherical waves	155
8.2.4. Expand an atomic orbital into plane waves	156
8.3. Overlap formula for atomic orbitals	157
8.4. Orbital overlap angular dependencies	159
8.5. Basis orthonormalization	162
8.5.1. Generalized eigenvalue problem	162
8.5.2. Löwdin symmetric orthogonalization	163

9. Re-centering method and multi-center matrix elements	167
9.1. Re-centering method	167
9.2. Overlap problem revisited	171
9.3. Hopping matrix elements	172
9.4. Long-range Coulomb matrix elements	177
9.5. Basis orthonormalization effects on the matrix elements	183
10. Summary	187
A. Hydrogen reference tables	189
A.1. Hydrogen-like radial wave functions	189
A.2. Slater-Condon parameters	191
A.3. Spin-orbit parameters	192
A.4. The first few spherical harmonics	193
A.5. The first few cubic harmonics	195
A.6. Orbital overlap angular dependences	197
B. Multiplet reference tables	199
B.1. Slater-Condon and Racah parameters	199
B.2. Table of open-shell multiplets	199
B.3. Hund's rule ground state term energies	201
B.4. Multiplet term energies	202
B.5. Multiplet and spin-orbit spectral variances	205

1. Many-electron states

1.1. Introducing a basis

Consider a potential created by six nuclei, forming a benzene-like ring (Fig. 1.1).

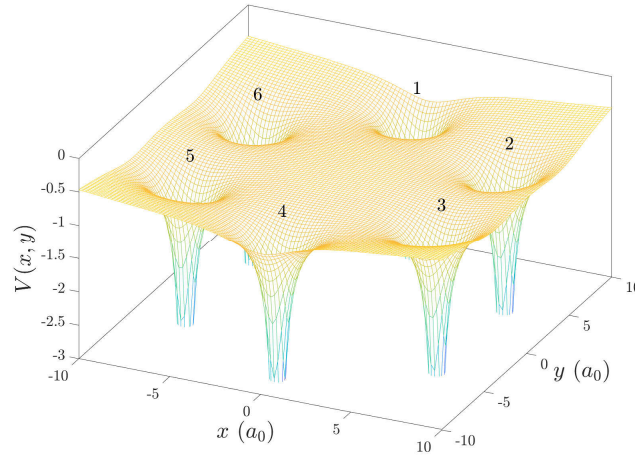


Figure 1.1.: A ring-shaped potential created by six nuclei.

By saying “given a potential created by nuclei”, we already implied the *Born-Oppenheimer approximation*, that the motion of electrons and nuclei can be separated. In the focus of describing electrons, the positions of the atomic nuclei are considered to be fixed.

Suppose the system contains a *single electron*. The one-electron Hamiltonian in atomic units [1] reads,

$$H = -\frac{1}{2}\nabla^2 - \sum_{\alpha=1}^6 \frac{Z_{\alpha}}{|\mathbf{r} - \mathbf{R}_{\alpha}|} \quad (1.1)$$

Encapsulating the electron-nuclei attraction as an external potential $V_{\text{ext}}(\mathbf{r})$, we can write down the one-electron Schrödinger equation:

$$\left[-\frac{1}{2}\nabla^2 + V_{\text{ext}}(\mathbf{r}) \right] \varphi(\mathbf{r}) = E\varphi(\mathbf{r}) \quad (1.2)$$

This is a textbook-like quantum mechanics problem. The external potential has a relatively complicated form, in the sense that Eqn. (1.2) cannot be solved analytically. However, by numerical methods, the one-electron problem is exactly solvable.

1. Many-electron states

Now, consider the system with *many electrons*. The many-electron Hamiltonian becomes,

$$H = \sum_{i=1}^{N_e} \left[-\frac{1}{2} \nabla_i^2 - \sum_{\alpha=1}^6 \frac{Z_\alpha}{|\mathbf{r}_i - \mathbf{R}_\alpha|} \right] + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \quad (1.3)$$

In addition to the one-body terms, the Hamiltonian has now $\binom{N_e}{2}$ pairs of two-body terms, namely, the electron-electron interaction. Denoting the nuclear potential as $V_{\text{ext}}(\mathbf{r}_i)$, we write down the many-electron Schrödinger equation:

$$\left\{ \sum_{i=1}^{N_e} \left[-\frac{1}{2} \nabla_i^2 + V_{\text{ext}}(\mathbf{r}_i) \right] + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \right\} \Psi(\mathbf{r}_1 \cdots \mathbf{r}_{N_e}) = E \Psi(\mathbf{r}_1 \cdots \mathbf{r}_{N_e}) \quad (1.4)$$

Comparing with (1.2), the difficulty of solving (1.4) blows up to a completely different level. The complication due to the external potential is in comparison almost negligible. The *real challenge* enters from two aspects:

1. To solve the problem numerically, one must represent the many-body wave function $\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_e})$ on a numerical grid. However, an N_e -electron wave function is $3N_e$ -dimensional. The number of grid points is “the number of grid points per dimension to the power of $3N_e$ ”.
2. The two-body term $\sum \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$ couples all the coordinates. If there were no such a term, the many-electron Schrödinger equation would be factorized completely to one-electron Schrödinger equations. The two-body term $\sum \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$ gives rise to the “many-body” problem.

The theory of (almost) everything:

In the most general problem setting, both electrons and the nuclei are described by the many-body wave function. The many-body Hamiltonian, without relativistic effects, is given by,

$$H = - \sum_i \frac{1}{2} \nabla_i^2 - \sum_\alpha \frac{1}{2M_\alpha} \nabla_\alpha^2 - \sum_{i,\alpha} \frac{Z_\alpha}{|\mathbf{r}_i - \mathbf{R}_\alpha|} + \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{\alpha < \beta} \frac{Z_\alpha Z_\beta}{|\mathbf{R}_\alpha - \mathbf{R}_\beta|} \quad (1.5)$$

Quoting Wigner and Seitz [2]:

If one had a great calculating machine, one might apply it to the problem of solving the Schrödinger equation for each metal and obtain thereby the interesting physical quantities, such as the cohesive energy, the lattice constant, and similar parameters. It is not clear, however, that a great deal would be gained by this. Presumably the results would agree with the experimentally determined quantities and nothing vastly new would be learned from the calculation. It would be preferable instead to have a vivid picture of the behavior of the wave functions, a simple description of the essence of

the factors which determine cohesion and an understanding of the origins of variation in properties [...].

The many-body Schrödinger equation is practically unsolvable. On the other hand, is it really necessary to have the exact solution of a many-body problem [3, 4]? It is often misleading to think that our job is to find approximations to the exact solution of the Schrödinger equation. Rather, the proper approach is to develop practical approximation schemes for explaining the main features of complicated systems. In most of the electronic structure calculations, we are interested in finding the ground state of the system. In the spirit of the *variational principle*, the one-electron basis orbitals are introduced.

Returning back to the *one-electron* problem. In our example, a “natural” choice of the basis orbitals are the six $1s$ -orbitals at each nucleus site. Denoting each $1s$ -orbital with its site number, we construct the approximated ground state:

$$|\text{gs}\rangle \approx a_1 |1s_1\rangle + a_2 |1s_2\rangle + a_3 |1s_3\rangle + a_4 |1s_4\rangle + a_5 |1s_5\rangle + a_6 |1s_6\rangle \quad (1.6)$$

If the nuclei are well separated (significantly greater than a Bohr radius), (1.6) would be a sufficiently good approximation. Eventually, in the limit of infinite separation, (1.6) becomes exact. However, when the nuclei are close, it is often overlooked, that while atomic orbitals are mutually orthogonal within a single atom, they are, in general, *non-orthogonal* for atoms on different lattice sites. When choosing a basis, orthonormal bases are primarily preferred. Because choosing a non-orthogonal basis will introduce a “generalized eigen-value problem” which involves additionally solving an overlap matrix and makes the problem unnecessarily complicated. In Chapter 8, we will discuss the orbital overlap problems and orbital orthogonalization in detail.

At this stage, we assume that the basis orbitals are orthogonal. The problem reduces from solving the differential equation to finding the six coefficients. For quantum chemists, this is the well known “LCAO” (linear combination of atomic orbitals) technique. In solid state physics, we call it the “tight-binding” method. The name is found in contrast to the “nearly-free electron model”, that instead of considering electrons in solids as plane waves, we consider electrons that are tightly bound to the atom to which they belong and they have limited interaction with states and potentials on surrounding atoms of the solid. The tight-binding method will be discussed in the next chapter.

In the case of *many-electron* systems, we need to work with many-electron states and consider the additional electron-electron interaction. There are two approaches to study the interaction:

1. A mean-field approach. We choose basis orbitals that are self-consistently constructed by mapping the electron-electron interaction to a mean-field potential. Effectively, we solve a one-electron problem. The self-consistent field (SCF) calculation for atomic orbitals will be discussed in Chapter 3.
2. A many-body approach. We construct many-body basis states and include explicitly the electron-electron interaction among individual electrons. In principle, this approach leads to exact solutions if the number of basis orbitals is infinite. However,

1. Many-electron states

we cannot afford to use a large number of basis orbitals, as the dimension of the many-body basis grows exponentially with the number of one-electron basis orbitals. The representation of the many-body basis will be discussed in the next section.

The second quantization [5] is formulated in the one-electron orbital basis, which provides a convenient way for handling many-body states. Given an (orthonormal) one-electron orbital basis, the exact Hamiltonian of an interacting system can be written in the form:

$$H = \underbrace{\sum_{\alpha\beta} h_{\alpha\beta} c_{\alpha}^{\dagger} c_{\beta}}_{\text{one-body}} + \underbrace{\sum_{\substack{\alpha<\beta \\ \gamma<\delta}} \ddot{v}_{\alpha\beta\gamma\delta} c_{\alpha}^{\dagger} c_{\beta}^{\dagger} c_{\gamma} c_{\delta}}_{\text{two-body}} \quad (1.7)$$

In the following sections in this chapter, we are going to explain the construction of the many-electron basis and the many-electron Hamiltonians in the language of second quantization.

Spins:

We were ignoring spins to simplify our discussions. However, to completely describe electrons, it is necessary to specify their spins. For two reasons: 1. There can be spin-dependent operators (e.g. the spin-orbit Hamiltonian); 2. Even in the absence of spin-dependent operators, spins are still necessary, as the many-electron wave function must be antisymmetric with respect to both spatial and spin coordinates.

In real-space, an electron is described by both the spatial coordinates $\mathbf{r}$ and the spin coordinate σ . Collectively, they are denoted by $\mathbf{x}$ [5, 6]:

$$\mathbf{x} = \{\mathbf{r}, \sigma\} \quad (1.8)$$

The proper notation for an N_e -electron wave function is therefore,

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_e}) \quad (1.9)$$

However, those troubles with coordinates vanish by working with basis orbitals. Because, given a basis, we only describe electrons by specifying which orbitals they occupy.

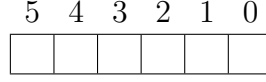
1.2. Representation of a many-electron basis

Given a basis, a one-electron state is a linear combination of one-electron basis states. Likewise, a many-electron state is a linear combination of many-electron basis states. How are the many-electron basis states constructed?

Consider a set of 6 one-electron basis orbitals (mutually orthonormal). The orbitals that we are discussing are *general*: they can be 6 orbitals at different sites (e.g. on a lattice),

1.2. Representation of a many-electron basis

or at one site (e.g. in an atom). The orbitals carry the spin quantum numbers, so one orbital can be occupied by at most one electron, *not two*. Each orbital has its own unique “ID”. For instance, each ID addresses the lattice site and the orbital quantum numbers (including spin). We label the 6 orbitals sequentially as “0, 1, 2, 3, 4, 5”. Pictorially, we can visualize them as 6 “boxes”:¹



Now, how many electrons do we have in the system? If the number of electrons is *fixed*, then we are working in the **Hilbert space**. If the number of electrons *varies*, ranging from 0 to 6 electrons, then we are working in the **Fock space**. Here, we discuss systems that have a fixed number of electrons, and provide a routine for setting up a many-electron basis in the Hilbert space (one can easily generalize the discussion to the Fock space).

Consider a system with 6 orbitals and 2 electrons. Putting 2 electrons into 6 orbitals is a combinatorics problem (electrons are identical, orbitals are distinguishable). There are “6-choose-2”

$$\binom{6}{2} = \frac{6 \times 5}{2!} = 15$$

possible ways. Explicitly, they are enumerated in Fig. 1.2. The 15 configurations represent the 15 many-electron basis states, which form the complete many-electron basis for the system with 6 orbitals and 2 electrons.

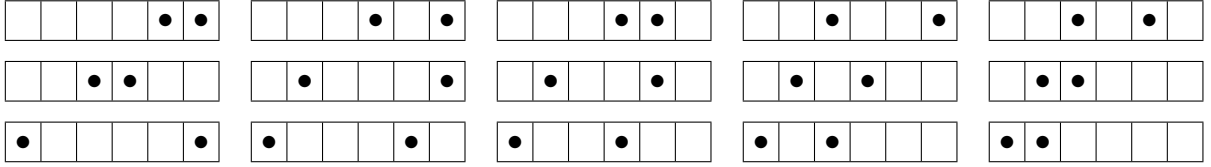


Figure 1.2.: The 15 configurations of a system with 6 orbitals and 2 electrons.

Electrons are fundamentally indistinguishable. Thus, a many-electron basis state does not contain the information of “which electron is in which orbital”. Rather, the only relevant information is “how many electrons are there in each orbital”. Therefore, a many-electron basis state can be precisely represented by a set of “occupation numbers”. Here we choose to represent a many-electron basis state in the order:

$$|n_5 \ n_4 \ n_3 \ n_2 \ n_1 \ n_0\rangle$$

For instance,

$$\left| \begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ \bullet & & & & \bullet & \end{array} \right\rangle \mapsto |100010\rangle \quad (1.10)$$

¹The order (from right to left) is chosen to be consistent with the convention of representing a “ket” state in the 2nd quantization: $c_5^\dagger c_4^\dagger c_3^\dagger c_2^\dagger c_1^\dagger c_0^\dagger |0\rangle$. It is also the natural order for the “bit representation” that the right-most digit is the “first” digit.

1. Many-electron states

Corresponding to Fig. 1.2, the occupation number representations read,

$$\begin{array}{ccccc} |000011\rangle & |000101\rangle & |000110\rangle & |001001\rangle & |001010\rangle \\ |001100\rangle & |010001\rangle & |010010\rangle & |010100\rangle & |011000\rangle \\ |100001\rangle & |100010\rangle & |100100\rangle & |101000\rangle & |110000\rangle \end{array}$$

One must have realized that they are simple binary numbers! Since electrons are *fermions*, the occupation numbers are either 0 or 1, thus the “occupation number representation” reduces to the “bit representation” of the many-electron states, which is extremely computer friendly. Now, generating this basis set is nothing but generating integer numbers. In this example, the corresponding integers in decimal format are

$$\begin{array}{ccccc} 3 & 5 & 6 & 9 & 10 \\ 12 & 17 & 18 & 20 & 24 \\ 33 & 34 & 36 & 40 & 48 \end{array}$$

How do we generate this set of integers? Here is a simple algorithm: Generate integers from 0 to 63 (in binary, that is from 000000 to 111111); for each integer, check if this integer in its binary format contains exactly two 1-bits; if yes, store this number. A simple Python (version 3) code for setting up a many-electron basis is given in Algorithm 1.1. This **Basis** class sets up the bit representation of the many-electron basis for a given number of orbitals N_{orb} and a given number of electrons N_e . Since we are working with binary numbers, we use bitwise operations, which are the most fundamental and the most efficient machine operations. Algorithm 1.2 provides a few common bit manipulation functions.

Algorithm 1.1: The **Basis** class (**basis.py**). A class using the bit representation for setting up a many-electron basis for given numbers of orbitals **Norb** and electrons **Ne**.

```

1 from bit import countBit
2
3 class Basis:
4     def __init__(self, Norb, Ne):
5         self.Norb = Norb # Number of orbitals
6         self.Ne = Ne # Number of electrons
7
8         # Basis configurations (e.g. conf[0]=000011, conf[7]=010010)
9         self.conf = [iconf for iconf in range(1<<Norb) if countBit(iconf)==Ne]
10
11        # Basis index dictionary (e.g. index[000011]=0, index[010010]=7)
12        self.index = {iconf: i for (i, iconf) in enumerate(self.conf)}
13
14        self.dim = len(self.conf) # Basis dimension (Norb-choose-Ne)

```

Algorithm 1.2: Bit manipulations (**bit.py**).

```

1 # Given n, count the number of 1's
2 def countBit(n):
3     return bin(n).count('1')
4

```

```

5  # Given n, test if the i-th bit is set
6  def isBit(n, i):
7      return bool(n & (1 << i))
8
9  # Given n, set the i-th bit
10 def setBit(n, i):
11     return n | (1 << i)
12
13 # Given n, clear the i-th bit
14 def clearBit(n, i):
15     return n & ~(1 << i)
16
17 # Given n, toggle the i-th bit
18 def toggleBit(n, i):
19     return n ^ (1 << i)

```

Counting the 1-bits:

In Algorithm 1.2, I used `bin(n).count('1')` to count the number of 1's in a binary number. It does the following: for a given number, say 34, `bin(34)` converts the integer into a binary string (prefixed with "0b") "100010". Then, the string calls the method `count('1')` to count the number of occurrences of "1". That is *inefficient*.

If you are familiar with C, there is a built-in function of GCC compiler:

```
int __builtin_popcount(unsigned int n)
```

It uses a processor instruction (if the hardware supports) that counts the 1-bit of `n`, which is as fast as one can get on a machine. In Python, one can use the `popcount` from the GMP library.

Alternatively, you do this:

```

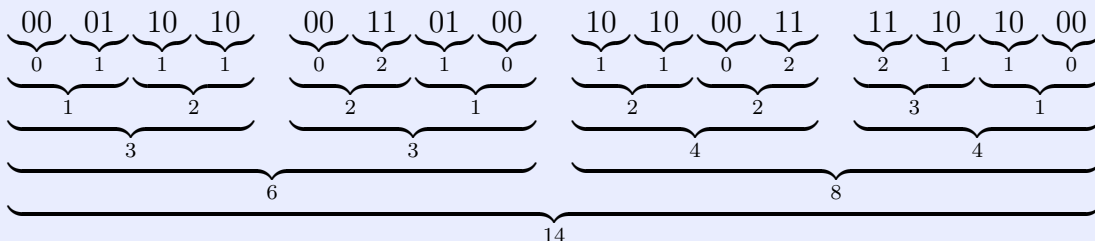
1  def countBit(n): # 0<=n<2**32
2      n = n - ((n >> 1) & 0x55555555)
3      n = (n & 0x33333333) + ((n >> 2) & 0x33333333)
4      return ((n + (n >> 4) & 0x0F0F0F0F) * 0x01010101) & 0xFFFFFFFF >> 24

```

But nobody understands this crazy code without explanation. This code works for any unsigned 32-bit integers. Suppose we have

$$n = 00011010\ 00110100\ 10100011\ 11101000$$

This algorithm counts 1-bits chunk-wise and in parallel (5 steps for a 32-bit integer):



Explicitly, the 5 steps are:

1. Many-electron states

```

1 def countBit(n): # 0<=n<2**32
2     n = (n&0x55555555) + ((n>> 1)&0x55555555)
3     n = (n&0x33333333) + ((n>> 2)&0x33333333)
4     n = (n&0x0F0F0F0F) + ((n>> 4)&0x0F0F0F0F)
5     n = (n&0x00FF00FF) + ((n>> 8)&0x00FF00FF)
6     n = (n&0x0000FFFF) + ((n>>16)&0x0000FFFF)
7     return n

```

With some arguments, one can show that those 5 steps can be simplified (and uglified) to the algorithm shown above (to push the algorithm to the extreme).

However, we are not focusing too much on the performance here. The priority is to explain how things can be implemented in principle. I would at the moment stay with `bin` and `count`, which is easy to read and works for an arbitrary-length integer.

The `Basis` class in Algorithm 1.1 is defined for setting up a general many-electron basis. It doesn't specify the physical meanings for each one-electron orbital. For example, when we discuss the *LS*-coupling and *jj*-coupling schemes in Chapter 4, we will define subclasses (Algorithms 4.1 and 4.3) that *inherit* from the `Basis` class. They additionally assign the one-electron orbitals with their physical meanings.

Given a many-electron basis, we can represent a general many-electron state in the basis as a *vector*. The vector lives in the Hilbert space, whose dimension is given by

$$\dim_{\text{H}} = \binom{N_{\text{orb}}}{N_e} \quad (1.11)$$

For large problem sizes, such a dimensionality makes it practically impossible to store a many-electron state. Table 1.1 shows the growth of the dimension in the Hilbert space for half-filled systems with N_{orb} number of orbitals. Simply by looking at the number of digits, we see that the growth of $\dim_{\text{H}}$ is exponential. Consider a Hubbard model with 20 sites with 2 orbitals per site (one for spin-up and one for spin-down), and the system is half-filled. That corresponds to a system with 40 orbitals and 20 electrons. For such a system, one needs a terabyte for storing a “single vector” in the double-precision floating point format.

Slater determinants:

If needed, we can express the many-body basis states in real-space as Slater determinants. For instance, (with a phase convention)

$$\left| \begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ \bullet & & & & \bullet & \end{array} \right\rangle \mapsto \Phi_{1,5}(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{\sqrt{2}} (\varphi_1(\mathbf{x}_1)\varphi_5(\mathbf{x}_2) - \varphi_5(\mathbf{x}_1)\varphi_1(\mathbf{x}_2)) \quad (1.12)$$

However, expressing a many-electron Slater determinant for more than two electrons will be “factorially” complicated due to the antisymmetrization with respect to the real-space coordinates.

Many-body states are most conveniently expressed in the second quantization lan-

1.3. Representation of one- and two-body Hamiltonians

guage. For the example above, we simply write (here the phase convention enters in the way that the orbital-index ordering has to be fixed)

$$\left| \begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ \bullet & & & & \bullet & \end{array} \right\rangle \mapsto c_5^\dagger c_1^\dagger |0\rangle \quad (1.13)$$

Note that the “bit representation” was introduced to *label* the many-body basis states. To actually *work* with the states, we use second quantization [5].

Table 1.1.: Dimension of Hilbert space for given N_{orb} and N_e . The column dim_H shows a fast growth in the logarithmic scale. The corresponding computer memories required for storing a “vector” (using double-precision) in the Hilbert space are given in the last column.

N_{orb}	N_e	dim_H	memory	
2	1	2	16	B
4	2	6	48	B
6	3	20	160	B
8	4	70	560	B
10	5	252	2	kB
12	6	924	7	kB
14	7	3432	27	kB
16	8	12870	103	kB
18	9	48620	389	kB
20	10	184756	1	MB
30	15	155117520	1	GB
40	20	137846528820	1	TB
50	25	126410606437752	1	PB

1.3. Representation of one- and two-body Hamiltonians

Hamiltonians (more generally speaking, hermitian operators) can be categorized by “the physical nature of the number of particles involved”. For example, in the many-electron Schrödinger equation (1.4):

1. The **one-body** operators are the kinetic Hamiltonian and the external potential Hamiltonian:

$$H_K = \sum_{i=1}^{N_e} -\frac{1}{2} \nabla_i^2 \quad \text{and} \quad H_{\text{ext}} = \sum_{i=1}^{N_e} V_{\text{ext}}(\mathbf{r}_i)$$

which operate on all *individual* electrons. If we extend our discussion to the spin-orbit interaction, the spin-orbit Hamiltonian is also a one-body operator (see (4.3)).

1. *Many-electron states*

2. The **two-body** operator is the Coulomb Hamiltonian:

$$H_U = \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$$

which operates on all *pairs* of electrons.

Consider a general one-body Hamiltonian H_1 and a general two-body Hamiltonian H_2 . In real-space (also called the first quantization), we write the general forms (including the spin coordinate):

$$H_1 = \sum_{i=1}^{N_e} h(\mathbf{x}_i) \quad \text{and} \quad H_2 = \sum_{i < j} v(\mathbf{x}_i, \mathbf{x}_j) \quad (1.14)$$

Because electrons are identical, there must be a restriction that v is symmetric in the arguments: $v(\mathbf{x}_i, \mathbf{x}_j) = v(\mathbf{x}_j, \mathbf{x}_i)$.

In second quantization, they are expressed as [5]:

$$\boxed{H_1 = \sum_{\alpha\beta} h_{\alpha\beta} c_{\alpha}^{\dagger} c_{\beta} \quad \text{and} \quad H_2 = \frac{1}{2} \sum_{\alpha\beta\gamma\delta} v_{\alpha\beta\gamma\delta} c_{\alpha}^{\dagger} c_{\beta}^{\dagger} c_{\gamma} c_{\delta}} \quad (1.15)$$

where $h_{\alpha\beta}$ and $v_{\alpha\beta\gamma\delta}$ are the corresponding matrix elements [5]:

$$h_{\alpha\beta} = \langle \alpha | h | \beta \rangle \quad \text{and} \quad v_{\alpha\beta\gamma\delta} = (\alpha\beta | v | \gamma\delta) \quad (1.16)$$

The one-electron Dirac notation $\langle \alpha | h | \beta \rangle$ is defined as

$$\langle \alpha | h | \beta \rangle \equiv \int d\mathbf{x} \overline{\varphi_{\alpha}}(\mathbf{x}) h(\mathbf{x}) \varphi_{\beta}(\mathbf{x}) \quad (1.17)$$

We define the two-electron integral notation $(\alpha\beta | v | \gamma\delta)$ using the “round bracket” as the matrix element in the basis of the *product* wave functions: (here enters a choice of ordering the coordinates)

$$(\alpha\beta | v | \gamma\delta) \equiv \int d\mathbf{x}_1 \int d\mathbf{x}_2 \overline{\varphi_{\alpha}}(\mathbf{x}_1) \overline{\varphi_{\beta}}(\mathbf{x}_2) v(\mathbf{x}_1, \mathbf{x}_2) \varphi_{\gamma}(\mathbf{x}_2) \varphi_{\delta}(\mathbf{x}_1) \quad (1.18)$$

However, we know that the product wave functions are not the proper two-electron basis functions. It is tempting to introduce the “real” two-electron integral with the “proper” basis functions – the two-electron Slater determinants:

$$\ddot{v}_{\alpha\beta\gamma\delta} = \langle \alpha\beta | v | \gamma\delta \rangle \quad (1.19)$$

The symbol $\ddot{v}$ is chosen with the hope that the “umlaut” reminds us that it is the matrix

1.3. Representation of one- and two-body Hamiltonians

element under the proper “two-electron” basis. Now,

$$\begin{aligned}
\langle \alpha\beta | v | \gamma\delta \rangle &\equiv \frac{1}{2} \int d\mathbf{x}_1 \int d\mathbf{x}_2 \\
&\quad \left[\overline{\varphi_\alpha}(\mathbf{x}_1) \overline{\varphi_\beta}(\mathbf{x}_2) - \overline{\varphi_\beta}(\mathbf{x}_1) \overline{\varphi_\alpha}(\mathbf{x}_2) \right] v(\mathbf{x}_1, \mathbf{x}_2) \left[\varphi_\delta(\mathbf{x}_1) \varphi_\gamma(\mathbf{x}_2) - \varphi_\gamma(\mathbf{x}_1) \varphi_\delta(\mathbf{x}_2) \right] \\
&= \frac{1}{2} \left[(\alpha\beta | v | \gamma\delta) - (\alpha\beta | v | \delta\gamma) - (\beta\alpha | v | \gamma\delta) + (\beta\alpha | v | \delta\gamma) \right] \\
&= \frac{1}{2} \left[(\alpha\beta | v | \gamma\delta) - (\alpha\beta | v | \delta\gamma) - (\alpha\beta | v | \delta\gamma) + (\alpha\beta | v | \gamma\delta) \right] \\
&= (\alpha\beta | v | \gamma\delta) - (\alpha\beta | v | \delta\gamma)
\end{aligned} \tag{1.20}$$

In short notations,

$$\boxed{\ddot{v}_{\alpha\beta\gamma\delta} = v_{\alpha\beta\gamma\delta} - v_{\alpha\beta\delta\gamma}} \tag{1.21}$$

Immediately, we see that $\ddot{v}_{\alpha\beta\gamma\delta}$ has the following symmetry properties:

$$\ddot{v}_{\alpha\beta\gamma\delta} = -\ddot{v}_{\alpha\beta\delta\gamma} = -\ddot{v}_{\beta\alpha\gamma\delta} = \ddot{v}_{\beta\alpha\delta\gamma} \tag{1.22}$$

Because electron-electron interactions are paired, it is actually a more “natural” choice to use the $\ddot{v}_{\alpha\beta\gamma\delta}$ elements, which are defined under the proper two-electron basis. A quick example is that,

$$\ddot{v}_{0012} = 0$$

The Pauli exclusion principle is internally built into the elements. On the other hand, v_{0012} (without umlaut) is in general a non-zero element (the Pauli exclusion principle is ensured by the operation $c_0^\dagger c_0^\dagger |\Psi\rangle = 0$).

By pairing the electrons, we can pair the indices (α, β) and (γ, δ) instead of enumerating over all the possibilities. For instance, we sort the indices as $(\alpha < \beta)$ and $(\gamma < \delta)$. The two-body Hamiltonian in (1.15) reduces to

$$\boxed{H_2 = \sum_{\substack{\alpha < \beta \\ \gamma < \delta}} \ddot{v}_{\alpha\beta\gamma\delta} c_\alpha^\dagger c_\beta^\dagger c_\gamma c_\delta} \tag{1.23}$$

Notice that pairing the indices significantly reduces the number of enumerations. For a system with N_{orb} orbitals, an exhaustive enumeration generates N_{orb}^4 tuples of indices. However, a large number of the indices can be considered as duplicates because of the symmetry, e.g. $(0, 1, 2, 3) = (1, 0, 3, 2)$; or unnecessary indices that do not contribute because of the Pauli principle, e.g. $(0, 0, 1, 2)$. Now, by introducing the pairing, the enumeration generates only the essential indices. There are $\binom{N_{\text{orb}}}{2}^2 = \frac{1}{4}N_{\text{orb}}^4 - \frac{1}{2}N_{\text{orb}}^3 + \frac{1}{4}N_{\text{orb}}^2$ of them.

Given a many-electron basis, the many-electron states are represented as *vectors*. Correspondingly, operators are represented as *matrices*. To set up the matrix representations $\mathbf{H}_1$ and $\mathbf{H}_2$, for each many-body basis state $|i\rangle$ and $|j\rangle$, we evaluate:

$$(\mathbf{H}_1)_{ij} = \langle i | H_1 | j \rangle = \sum_{\alpha\beta} h_{\alpha\beta} \langle i | c_\alpha^\dagger c_\beta | j \rangle \tag{1.24}$$

$$(\mathbf{H}_2)_{ij} = \langle i | H_2 | j \rangle = \sum_{\substack{\alpha < \beta \\ \gamma < \delta}} \ddot{v}_{\alpha\beta\gamma\delta} \langle i | c_\alpha^\dagger c_\beta^\dagger c_\gamma c_\delta | j \rangle \tag{1.25}$$

1. Many-electron states

Suppose the matrix elements $h_{\alpha\beta}$ (input **h**) and $\ddot{v}_{\alpha\beta\gamma\delta}$ (input **vee**) are prepared, the Python routines for setting up $\mathbf{H}_1$ and $\mathbf{H}_2$ are shown in Algorithm (1.3).

Algorithm 1.3: Matrix representations of a one-body Hamiltonian H_1 and a two-body Hamiltonian H_2 in a many-electron basis (**hamiltonian.py**).

```

1 from basis import Basis
2 from bit import countBit, isBit, setBit, clearBit
3 import numpy as np
4
5 # Matrix representation of a one-body Hamiltonian
6 def H1(Norb, Ne, h):
7     # Set up basis
8     B = Basis(Norb, Ne)
9
10    # Orbital indices
11    idx = [(a,b) for a in range(Norb) for b in range(Norb)]
12
13    # Set up the matrix representation
14    M = np.zeros((B.dim,B.dim))
15    for i in range(B.dim):           # i (index)
16        iconf = B.conf[i]           # conf <i|
17        for (a,b) in idx:
18            if isBit(iconf, a):
19                aconf = clearBit(iconf, a)   # conf <i|a'
20                if not isBit(aconf, b):
21                    bconf = setBit(aconf, b) # conf <i|a'b = conf |j>
22                    j = B.index[bconf]      # j (index)
23                    # Fermi-sign
24                    account = countBit(aconf&(-1<<(a+1)))
25                    bcount = countBit(bconf&(-1<<(b+1)))
26                    fsign = 1-2*((account+bcount)&1)
27                    M[i,j] += fsign*h[a,b]
28
29    return M
30
31 # Matrix representation of a two-body Hamiltonian
32 def H2(Norb, Ne, vee):
33     # Set up basis
34     B = Basis(Norb, Ne)
35
36     # Orbital indices
37     idx = [(a,b,c,d) for a in range(Norb) for b in range(Norb) if a<b
38              for c in range(Norb) for d in range(Norb) if c<d]
39
40    # Set up the matrix representation
41    M = np.zeros((B.dim,B.dim))
42    for i in range(B.dim):           # i (index)
43        iconf = B.conf[i]           # conf <i|
44        for (a,b,c,d) in idx:
45            if isBit(iconf, a):
46                aconf = clearBit(iconf, a)   # conf <i|a'
47                if isBit(aconf, b):
48                    bconf = clearBit(aconf, b) # conf <i|a'b'
49                    if not isBit(bconf, c):
50                        cconf = setBit(bconf, c) # conf <i|a'b'c
51                        if not isBit(ccconf, d):
52                            dconf = setBit(ccconf, d) # conf <i|a'b'c d = conf |j>
53                            j = B.index[dconf]      # j (index)
54                            # Fermi-sign
55                            account = countBit(aconf&(-1<<(a+1)))
56                            bcount = countBit(bconf&(-1<<(b+1)))
57                            cccount = countBit(ccconf&(-1<<(c+1)))
58                            dcount = countBit(dconf&(-1<<(d+1)))
59                            fsign = 1-2*((account+bcount+cccount+dcount)&1)
60                            M[i,j] += fsign*vee[a,b,c,d]
61
62    return M

```

Fermi sign:

When applying creators or annihilators to a state, one must sort the operators in the “correct” order (phase convention). Because the operators anti-commute, a Fermi sign (± 1) will be generated.

The complete “algebra” of second quantization is described as [5]:

$$\begin{aligned} \langle 0 | 0 \rangle &= 1 \\ c_\alpha | 0 \rangle &= 0 \\ \{c_\alpha^\dagger, c_\beta^\dagger\} &= 0 \\ \{c_\alpha, c_\beta\} &= 0 \\ \{c_\alpha, c_\beta^\dagger\} &= \langle \alpha | \beta \rangle \end{aligned} \tag{1.26}$$

Now, consider applying a creator $c_2^\dagger$ to a state $c_6^\dagger c_4^\dagger c_3^\dagger c_0^\dagger | 0 \rangle$. Our convention is to sort the orbital indices in an ascending order from right to left. Thus, we need to anti-commute $c_2^\dagger$ with $c_6^\dagger$, $c_4^\dagger$, and $c_3^\dagger$ (swap 3 times). The resulting Fermi sign is $(-1)^3 = -1$. Explicitly,

$$c_2^\dagger c_6^\dagger c_4^\dagger c_3^\dagger c_0^\dagger | 0 \rangle = (-1)^3 c_6^\dagger c_4^\dagger c_3^\dagger c_2^\dagger c_0^\dagger | 0 \rangle$$

How is this implemented? In the bit representation, we need to count the number of 1-bits “in front of orbital-2”. We can achieve this using a **mask**:

```
# 6543210
conf = 1011101
mask = 1111000
count = countBit(conf&mask)
```

where the **mask** can be created by shifting a bunch of 1’s to the left: $-1 \ll (i+1)$ (-1 is represented by “all-bit ones” in *two’s-complement representation*), where **i** is the orbital index (**i**=2 in the example).

To get the Fermi sign, it is a matter of taking (-1) to the power of the **count**. One can optimize this operation using a binary operation:

```
fsign = 1-2*(count&1)
```

where **(count&1)** returns 1 if **count** is odd, or 0 if **count** is even.

To set up the matrix representations, the main work is the evaluations of $\langle i | c_\alpha^\dagger c_\beta | j \rangle$ and $\langle i | c_\alpha^\dagger c_\beta^\dagger c_\gamma c_\delta | j \rangle$. The operations are basically taking the inner product of two many-electron states $\langle I | J \rangle$. It is just a matter of perspective to view $\langle I |$ and $| J \rangle$, for instance,

$$\begin{cases} \langle I | \leftarrow \langle i | c_\alpha^\dagger c_\beta^\dagger c_\gamma c_\delta \\ | J \rangle \leftarrow | j \rangle \end{cases} \quad \text{or} \quad \begin{cases} \langle I | \leftarrow \langle i | c_\alpha^\dagger c_\beta^\dagger \\ | J \rangle \leftarrow c_\gamma c_\delta | j \rangle \end{cases}$$

1. Many-electron states

In either case, $\langle I|$ and $|J\rangle$ are states with the same number of electrons. Given the “rules” in (1.26) and the assumption that the one-electron orbitals are orthonormal, it follows that $\langle I|$ and $|J\rangle$ must have the same configuration, such that $\langle I|J\rangle \neq 0$. If $\langle I|$ and $|J\rangle$ have the same configuration, their inner product is simply a Fermi sign (± 1).

The routines in Algorithm 1.3 are general for setting up the matrix representations for one- and two-body Hamiltonians in a many-electron basis. To be system specific, the physical information enters via the matrix elements $h_{\alpha\beta}$ and $\ddot{v}_{\alpha\beta\gamma\delta}$. In Chapter 5, we are going to investigate trends of the matrix elements for the spin-orbit Hamiltonian (one-body) and the Coulomb Hamiltonian (two-body) for realistic atomic basis orbitals.

Because the Hilbert space dimension is in general huge, the corresponding matrix representations are “huge-squared”. But, the matrix representations are, in general, very sparse. The reason of the “sparseness” is the following: Consider the operation $\langle i|c_{\alpha}^{\dagger}c_{\beta}^{\dagger}c_{\gamma}c_{\delta}|j\rangle$. It is *always zero* if the configurations of $\langle i|$ and $|j\rangle$ differ by three electrons (or more). For instance,

$$\left(\langle 0|c_1c_2c_3\right)c_{\alpha}^{\dagger}c_{\beta}^{\dagger}c_{\gamma}c_{\delta}\left(c_6^{\dagger}c_5^{\dagger}c_4^{\dagger}|0\rangle\right)=0 \quad \forall (\alpha, \beta, \gamma, \delta)$$

since $\langle 0|c_1c_2c_3$ and $c_6^{\dagger}c_5^{\dagger}c_4^{\dagger}|0\rangle$ differ by three electrons, there is no $(\alpha, \beta, \gamma, \delta)$ to bring the two to the same configuration.

Methods like the Lanczos [7] are suited for dealing with large sparse Hamiltonians. For relatively small systems, we can set up the sparse matrix representations and perform a Lanczos diagonalization. The restrictions are the available computer memory. Due to the huge memory consumption of exact diagonalization methods, it can be understood that various methods have been developed to tackle many-body problems from different perspectives. However, there exist precious systems in which the many-body Hamiltonians can be solved *analytically*. In Chapter 4, we discuss the atomic open-shell problems, where the systems can be solved analytically (up to seniority) without setting up a matrix representation. In Chapter 6, we discuss an analytical approach to compute *moments* of general one- and two-body Hamiltonians.

2. The tight-binding method

2.1. The tight-binding Hamiltonian

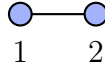
We have discussed in the beginning of Chapter 1: while one-body Hamiltonians are exactly solvable, two-body Hamiltonians give rise to the real “many-body” challenges. The tight-binding Hamiltonian is simply *one-body*:

$$H = \sum_{\alpha\beta} (-t_{\alpha\beta}) c_{\alpha}^{\dagger} c_{\beta} \quad (2.1)$$

How about the missing two-body term that describes the electron-electron interaction? This is to be understood in a *mean-field* picture: the complicated electron-electron interaction is treated effectively as a single electron interacting with a mean-field potential. In a mean-field approximation, the two-body term reduces to a one-body term plus some “two-body leftover”. In the tight-binding method, the “two-body leftover” is dropped, or it can be treated later as a perturbation.¹

Being an effectively non-interacting system (a one-electron problem), the tight-binding Hamiltonian is easy to solve. The nightmare with the huge Hilbert space dimension $\dim = \binom{N_{\text{orb}}}{N_e}$ collapses to $\dim = N_{\text{orb}}$. Tight-binding method is practical, and it is typically used for describing band structures in solids where the electron correlations are less important.

Now, let’s take a close look at the matrix element $(-t_{\alpha\beta})$. This can be easily illustrated using a simple two-site model: consider a hydrogen molecule ion H_2^+ , which consists of a single electron and two protons (site 1 and 2):



The one-electron Hamiltonian reads,

$$H = -\frac{1}{2}\nabla^2 + V_1(\mathbf{r}) + V_2(\mathbf{r}) \quad (2.2)$$

Since this is a one-electron problem, and the Hamiltonian does not involve spins, the up-spin and down-spin spaces simply factorize. To set up the one-electron basis, we only

¹The two-body terms are crucial for strongly correlated systems where the two-body interactions are necessary for describing the materials. This class of materials includes many transition metal oxides.

2. The tight-binding method

need to consider orbitals with the same spin. Thus, we write down,

$$|\text{gs}\rangle = a_1 |1s_1\rangle + a_2 |1s_2\rangle \mapsto \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad (2.3)$$

In the basis, there are two different kinds of matrix elements:

$$-t_{11} = \langle 1s_1 | H | 1s_1 \rangle \quad \text{and} \quad -t_{12} = \langle 1s_1 | H | 1s_2 \rangle$$

The first one is called the *on-site* energy:

$$-t_{11} = \langle 1s_1 | -\frac{1}{2}\nabla^2 + V_1 + V_2 | 1s_1 \rangle = \varepsilon_{1s} + \langle 1s_1 | V_2 | 1s_1 \rangle \quad (2.4)$$

If the two sites are well separated, the on-site energy ($-t_{11}$) essentially converges to ε_{1s} .

The second one is called the *hopping* matrix element:

$$-t_{12} = \langle 1s_1 | -\frac{1}{2}\nabla^2 + V_1 + V_2 | 1s_2 \rangle = \varepsilon_{1s} \langle 1s_1 | 1s_2 \rangle + \langle 1s_1 | V_1 | 1s_2 \rangle \quad (2.5)$$

If the two sites are well separated, both the overlap $\langle 1s_1 | 1s_2 \rangle$ and the integral $\langle 1s_1 | V_1 | 1s_2 \rangle$ become very small due to the exponential decaying behavior of the wave functions. The hopping matrix element ($-t_{12}$) drops quickly as a function of site distance. That is why one often makes the simplification (on a lattice) that “an electron only hops to its nearest neighbors”. We can pretty well let the electron hop to further neighbors, but just with a much weaker hopping matrix element.

I haven’t explained why the matrix element ($-t_{\alpha\beta}$) carries a mysterious minus sign. To explain that, we need to diagonalize the Hamiltonian. At this moment, we assume that the two basis orbitals are orthonormal, so that we have a simple eigen-value problem instead of a generalized one. Now, the matrix representation of the Hamiltonian:

$$\mathbf{H} = \begin{bmatrix} -t_{11} & -t_{12} \\ -t_{21} & -t_{22} \end{bmatrix} = \begin{bmatrix} X & Y \\ Y & X \end{bmatrix} \quad (2.6)$$

This matrix has two eigen-states:

$$E_1 = X + Y, \quad \mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (2.7)$$

$$E_2 = X - Y, \quad \mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (2.8)$$

Here comes the *physics*: we know that a wave function with more “nodes” corresponds to a higher energy state (this is something one can’t explain with pure algebra). Here $\mathbf{v}_1$ has zero node but $\mathbf{v}_2$ has one node, it must be that $E_2 > E_1$, which means Y is negative. In other words, $-t_{12}$ and $-t_{21}$ are negative. So the hopping matrix elements come with negative signs by the nature of physics.

2.2. Bloch's theorem

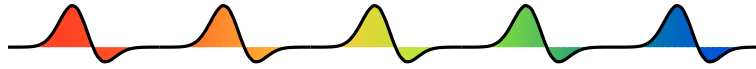
H_2^+ was simple enough. But if we are talking about real materials that contain an Avogadro number of sites, even for a one-electron system, the basis is too large. Here comes Bloch's theorem that vastly simplifies the problem when a periodic potential is presented. The idea is, if we have a unit cell which repeats periodically,



then the solution should also repeat periodically from cell to cell:



Well, not completely true. The wave function repeats periodically up to a phase (a unit complex constant):



Now, we go through Bloch's theorem more mathematically: Consider a periodic potential

$$V(\mathbf{r} + \mathbf{a}) = V(\mathbf{r}) \quad (2.9)$$

Let $T(\mathbf{a})$ be the translation operator

$$T(\mathbf{a}) |\varphi(\mathbf{r})\rangle = |\varphi(\mathbf{r} + \mathbf{a})\rangle \quad (2.10)$$

The eigen-energy of the system shouldn't be changed (while the wave function will be changed by a phase) under a translation $\mathbf{a}$:

$$\langle \varphi(\mathbf{r}) | H | \varphi(\mathbf{r}) \rangle = E = \langle \varphi(\mathbf{r} + \mathbf{a}) | H | \varphi(\mathbf{r} + \mathbf{a}) \rangle \quad (2.11)$$

which means,

$$H = T^\dagger(\mathbf{a}) H T(\mathbf{a}) \quad (2.12)$$

Eqn. (2.12) is equivalent to the relation that the two operators commute

$$[T(\mathbf{a}), H] = 0 \quad (2.13)$$

Therefore, if we know the eigen-functions of $T(\mathbf{a})$, then we know the eigen-functions of H . So, what are the eigen-functions of $T(\mathbf{a})$? They are:

$$\varphi_{\mathbf{k}}(\mathbf{r}) = \underbrace{e^{i\mathbf{k} \cdot \mathbf{r}}}_{\text{phase}} \underbrace{u(\mathbf{r})}_{\text{periodic}} \quad (2.14)$$

where $u(\mathbf{r} + \mathbf{a}) = u(\mathbf{r})$. Pictorially,

2. The tight-binding method

$$e^{i\mathbf{k}\cdot\mathbf{r}} = \text{[color bar from red to blue]}$$

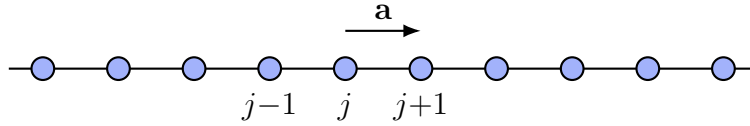
$$u(\mathbf{r}) = \text{[periodic wave function plot]}$$

Function (2.14) is called the Bloch wave. It consists of a plane wave phase component and a periodic function with the same periodicity as the lattice structure. We can easily confirm that the Bloch waves are indeed the eigen-functions of the translation operator:

$$T(\mathbf{a})\varphi_{\mathbf{k}}(\mathbf{r}) = T(\mathbf{a})e^{i\mathbf{k}\cdot\mathbf{r}}u(\mathbf{r}) = e^{i\mathbf{k}\cdot(\mathbf{r}+\mathbf{a})}u(\mathbf{r}+\mathbf{a}) = e^{i\mathbf{k}\cdot\mathbf{a}}e^{i\mathbf{k}\cdot\mathbf{r}}u(\mathbf{r}) = \underbrace{e^{i\mathbf{k}\cdot\mathbf{a}}}_{\text{const.}}\varphi_{\mathbf{k}}(\mathbf{r}) \quad (2.15)$$

2.3. One-dimensional lattice problem

We are now in a position to solve the tight-binding model on the simplest lattice: a 1D chain of repeating hydrogen nuclei. This chain has an infinite number of sites. At each site, we only consider an $1s$ orbital of the same spin. In the entire system, only one electron is presented.



Here comes the power of Bloch's theorem: we can easily construct the eigen-states of the periodic system. The first step, we write down the “periodic component”, which is a simple summation of all basis orbitals: (in this case $N = \infty$, infinite system size, not normalizable)

$$u(\mathbf{r}) \mapsto \frac{1}{\sqrt{N}} \sum_j c_j^\dagger |0\rangle \quad (2.16)$$

The second step is to add the “phase components” onto each site:

$$\varphi_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}}u(\mathbf{r}) \mapsto \frac{1}{\sqrt{N}} \sum_j e^{i\mathbf{k}\cdot\mathbf{r}_j} c_j^\dagger |0\rangle = c_{\mathbf{k}}^\dagger |0\rangle \quad (2.17)$$

which is the Bloch wave represented in an orbital basis. Now, we shall confirm that $c_{\mathbf{k}}^\dagger |0\rangle$ are indeed the eigen-states of the system. Because of the translational symmetry, we can write the tight-binding Hamiltonian in the form:

$$H = \sum_{j\xi} (-t_\xi) c_{j+\xi}^\dagger c_j \quad (2.18)$$

where j is the site index, and ξ denotes j 's neighbor. Notice that $(-t_\xi)$ is independent of j , but only depends on how far the neighbor is. Now,

$$H c_{\mathbf{k}}^\dagger |0\rangle = \left(\sum_{j'\xi} (-t_\xi) c_{j'+\xi}^\dagger c_{j'} \right) \left(\frac{1}{\sqrt{N}} \sum_j e^{i\mathbf{k}\cdot\mathbf{r}_j} c_j^\dagger |0\rangle \right) = \frac{1}{\sqrt{N}} \sum_{j\xi} (-t_\xi e^{i\mathbf{k}\cdot\mathbf{r}_j}) c_{j+\xi}^\dagger |0\rangle \quad (2.19)$$

Re-labelling the index $j \leftarrow j + \xi$, we obtain,

$$H c_{\mathbf{k}}^{\dagger} |0\rangle = \frac{1}{\sqrt{N}} \sum_{j\xi} (-t_{\xi} e^{i\mathbf{k} \cdot (\mathbf{r}_j - \mathbf{r}_{\xi})}) c_j^{\dagger} |0\rangle = \underbrace{\sum_{\xi} (-t_{\xi} e^{-i\mathbf{k} \cdot \mathbf{r}_{\xi}})}_{E_{\mathbf{k}}} \underbrace{\left(\frac{1}{\sqrt{N}} \sum_j e^{i\mathbf{k} \cdot \mathbf{r}_j} c_j^{\dagger} |0\rangle \right)}_{c_{\mathbf{k}}^{\dagger} |0\rangle} \quad (2.20)$$

Evidently, $c_{\mathbf{k}}^{\dagger} |0\rangle$ is an eigen-state of H with the eigen-energy

$$E_{\mathbf{k}} = \sum_{\xi} (-t_{\xi} e^{-i\mathbf{k} \cdot \mathbf{r}_{\xi}}) \quad (2.21)$$

The above result is also true for lattices with higher dimensions. In this example, we simply have:

$$\mathbf{k} \cdot \mathbf{r}_{\xi} = \mathbf{k} \cdot (\xi \mathbf{a}) = \xi k a$$

Let's say the matrix elements have the form

$$-t_{\xi} = \begin{cases} \varepsilon & \text{if } \xi = 0 \\ -t & \text{if } \xi = \pm 1 \\ -t' & \text{if } \xi = \pm 2 \\ 0 & \text{otherwise} \end{cases} \quad (2.22)$$

Accordingly, the eigen-energy reads: (plotted in Fig. 2.1)

$$E_k = \sum_{\xi=-2}^2 (-t_{\xi} e^{-i\xi k a}) = \varepsilon - 2t \cos(ka) - 2t' \cos(2ka) \quad (2.23)$$

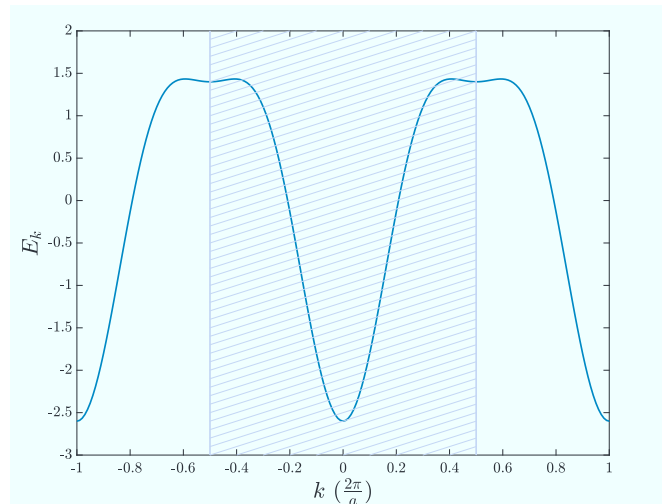


Figure 2.1.: Energy band of a one-dimensional lattice with parameters $\varepsilon = 0$, $t = 1$, $t' = 0.3$. Dashed interval: the first Brillouin zone.

2. The tight-binding method

Apparently, the energy band in Fig. 2.1 has a period $\frac{2\pi}{a}$. One period of the $\mathbf{k}$ region is called a Brillouin zone. Within a Brillouin zone, all $\mathbf{k}$ vectors correspond to distinct eigenstates. However, once moving out a Brillouin zone, we start to reproduce old eigenstates. That is in fact a bit counter-intuitive: larger magnitude in $\mathbf{k}$ means higher momentum and thus should point to a higher energy state. That is true for continuous space. On a lattice, a low momentum state can be the same as a high momentum state. How is that possible? If we compare the two waves in the *complete real space*, they are indeed different (see Fig. 2.2). But if we only look at wave functions at the *grid points*, the oscillation *between* two grid points does not contribute.

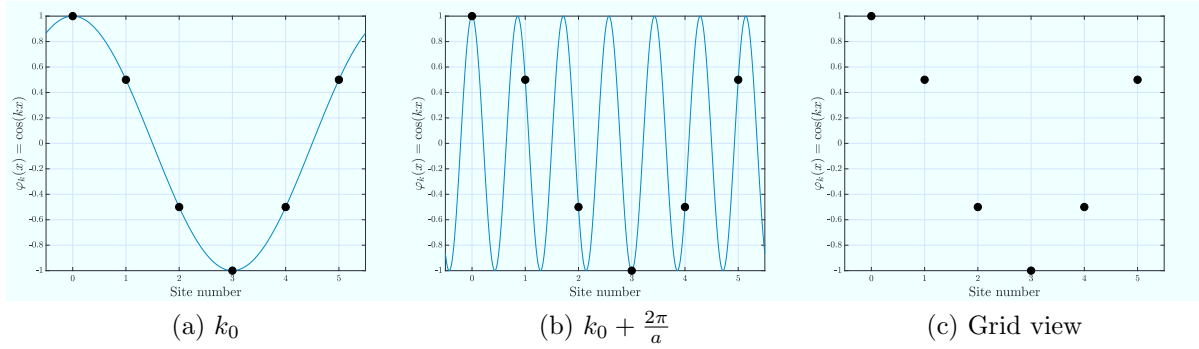


Figure 2.2.: Two Bloch waves with momentum k_0 and $k_0 + \frac{2\pi}{a}$, respectively. They represent the same state on the grid.

In our infinite lattice chain problem, all continuous k values are allowed. Very often people will impose a periodic boundary condition, which says, the chain has a finite number of sites; and at the “ $N + 1$ ” site, the wave function must have the same value as on the first site. This is like wrapping the chain around to form a ring. This new problem is almost the same as the chain with infinite sites. The only difference is that, due to the boundary condition, not all k ’s are allowed, they are now quantized. Because we need to match the phase on the connecting points:

$$e^{ik(N+1)a} = e^{ika} \Rightarrow kNa = 2\pi n \quad (2.24)$$

Within a Brillouin zone $[0, \frac{2\pi}{a})$, the allowed values are

$$k = 0 \times \frac{2\pi}{Na}, 1 \times \frac{2\pi}{Na}, 2 \times \frac{2\pi}{Na}, \dots, (N-1) \times \frac{2\pi}{Na}$$

Attention: do not confuse the two concepts

- Periodic potential: Wave functions match up to a phase.
- Periodic boundary condition: Wave functions match completely.

I would like to revise our problem from the matrix-vector point of view. To work with matrices and vectors, we only solve problems with finite sizes. The 6-site ring system (Fig. 1.1) that we discussed in the beginning of Chapter 1 would be an ideal candidate.

2.4. Multi-band one-dimensional lattice problem

Assuming the electron hops up to the second nearest neighbor, the matrix representation of the Hamiltonian reads,

$$\mathbf{H} = \begin{array}{c|cccccc} H & |1s_1\rangle & |1s_2\rangle & |1s_3\rangle & |1s_4\rangle & |1s_5\rangle & |1s_6\rangle \\ \hline \langle 1s_1| & \varepsilon & -t & -t' & & -t' & -t \\ \langle 1s_2| & -t & \varepsilon & -t & -t' & & -t' \\ \langle 1s_3| & -t' & -t & \varepsilon & -t & -t' & \\ \langle 1s_4| & & -t' & -t & \varepsilon & -t & -t' \\ \langle 1s_5| & -t' & & -t' & -t & \varepsilon & -t \\ \langle 1s_6| & -t & -t' & & -t' & -t & \varepsilon \end{array} \quad (2.25)$$

If we came across such a matrix and we didn't know the physics, most likely we would throw the matrix to an eigen-solver and obtain results numerically. Now, with Bloch's theorem, we immediately write down the eigen-vectors,

$$\mathbf{v}_k = \frac{1}{\sqrt{6}} \begin{bmatrix} e^{ik0a} \\ e^{ik1a} \\ e^{ik2a} \\ e^{ik3a} \\ e^{ik4a} \\ e^{ik5a} \end{bmatrix} \quad (2.26)$$

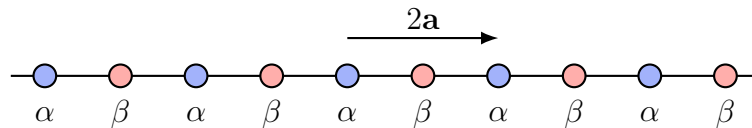
From a simple matrix-vector multiplication, we get (using B.C. $\Rightarrow e^{ik5a} = e^{-ik1a}$, etc.)

$$\mathbf{H}\mathbf{v}_k = \frac{1}{\sqrt{6}} \begin{bmatrix} \varepsilon e^{ik0a} - t(e^{ik1a} + e^{ik5a}) - t'(e^{ik2a} + e^{ik4a}) \\ \varepsilon e^{ik1a} - t(e^{ik2a} + e^{ik0a}) - t'(e^{ik3a} + e^{ik5a}) \\ \varepsilon e^{ik2a} - t(e^{ik3a} + e^{ik1a}) - t'(e^{ik4a} + e^{ik0a}) \\ \varepsilon e^{ik3a} - t(e^{ik4a} + e^{ik2a}) - t'(e^{ik5a} + e^{ik1a}) \\ \varepsilon e^{ik4a} - t(e^{ik5a} + e^{ik3a}) - t'(e^{ik0a} + e^{ik2a}) \\ \varepsilon e^{ik5a} - t(e^{ik0a} + e^{ik4a}) - t'(e^{ik1a} + e^{ik3a}) \end{bmatrix} = [\varepsilon - 2t \cos(ka) - 2t' \cos(2ka)] \mathbf{v}_k \quad (2.27)$$

which agrees with the result in (2.23).

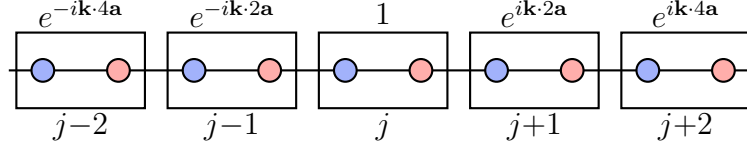
2.4. Multi-band one-dimensional lattice problem

How does Bloch's theorem work if one unit cell contains many orbitals? Consider a variation of the one-dimensional lattice:



It is similar to the one-dimensional lattice problem that we have discussed, but on every second site, the nucleus α is replaced by nucleus β . On this new lattice, the primitive vector is no longer $\mathbf{a}$, but $2\mathbf{a}$. A unit cell groups one α and one β together.

2. The tight-binding method



For this lattice system, we have Bloch states for α and β orbitals, respectively,

$$c_{\mathbf{k}\alpha}^\dagger |0\rangle = \frac{1}{\sqrt{N}} \sum_j e^{i\mathbf{k} \cdot \mathbf{r}_j} c_{j\alpha}^\dagger |0\rangle \quad (2.28)$$

$$c_{\mathbf{k}\beta}^\dagger |0\rangle = \frac{1}{\sqrt{N}} \sum_j e^{i\mathbf{k} \cdot \mathbf{r}_j} c_{j\beta}^\dagger |0\rangle \quad (2.29)$$

They are not the eigen-states, but they form a (2-dimensional) basis of the system. Now we construct the matrix representation of the Hamiltonian in the basis. Here we assume the hopping element between adjacent α - β is $(-t)$; the hopping elements between two nearest α - α and β - β are $(-t'_\alpha)$ and $(-t'_\beta)$, respectively. Further hoppings are ignored. The matrix representation reads,

$$\mathbf{H} = \begin{bmatrix} \varepsilon_\alpha - t'_\alpha (e^{-2ika} + e^{2ika}) & -t (1 + e^{-2ika}) \\ -t (1 + e^{2ika}) & \varepsilon_\beta - t'_\beta (e^{-2ika} + e^{2ika}) \end{bmatrix} \quad (2.30)$$

Diagonalizing the matrix, we obtain two energy bands:

$$E_k = \frac{1}{2} \left[A + B \pm \sqrt{(A - B)^2 + 4C} \right] \quad (2.31)$$

where,

$$\begin{aligned} A &= \varepsilon_\alpha - 2t'_\alpha \cos(2ka) \\ B &= \varepsilon_\beta - 2t'_\beta \cos(2ka) \\ C &= t^2(2 + 2\cos(2ka)) \end{aligned}$$

Now we put some numbers to see the behavior of the two energy bands. Let's say,

$$\underbrace{\varepsilon_\alpha = 0, \varepsilon_\beta = 0.5}_{\text{on site}}, \quad \underbrace{t = 1}_{\text{1st neighbor}}, \quad \underbrace{t'_\alpha = 0.2, t'_\beta = 0.4}_{\text{2nd neighbor}}$$

The corresponding energy bands are plotted in Fig. 2.3. We have two bands since there are two orbitals per unit cell. Now you might ask, what happens if α and β were actually the same nuclei? Let's say,

$$\underbrace{\varepsilon_\alpha = \varepsilon_\beta = 0}_{\text{on site}}, \quad \underbrace{t = 1}_{\text{1st neighbor}}, \quad \underbrace{t'_\alpha = t'_\beta = 0.3}_{\text{2nd neighbor}}$$

This actually recovers the one-dimensional lattice problem that we discussed in the previous section. And the corresponding two energy bands are plotted in Fig. 2.4. However, there is a subtle difference in the *interpretation*. The point is, since we grouped α, β to one cell, the “lattice” that we are referring is coarser. The Brillouin zone has a range $[0, \frac{\pi}{a})$, which is as half as before. But now for each $\mathbf{k}$, we have two states. So the number of states is doubled while the Brillouin zone is halved.

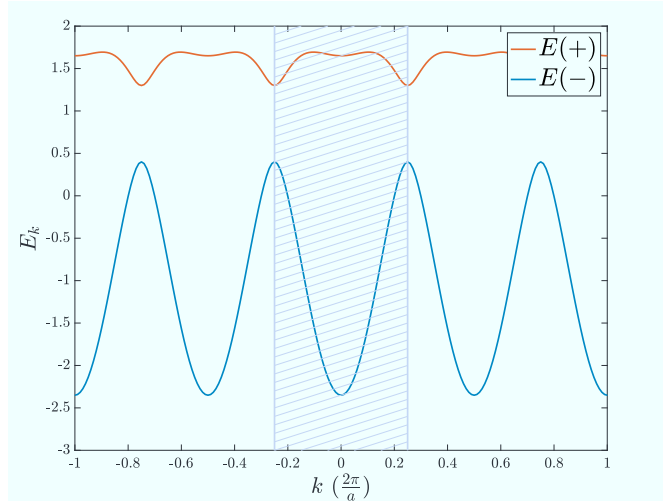


Figure 2.3.: Energy bands of a one-dimensional $\alpha\beta\alpha$ -chain with parameters $\varepsilon_\alpha = 0$, $\varepsilon_\beta = 0.5$, $t = 1$, $t'_\alpha = 0.2$, $t'_\beta = 0.4$. Dashed interval: the first Brillouin zone.

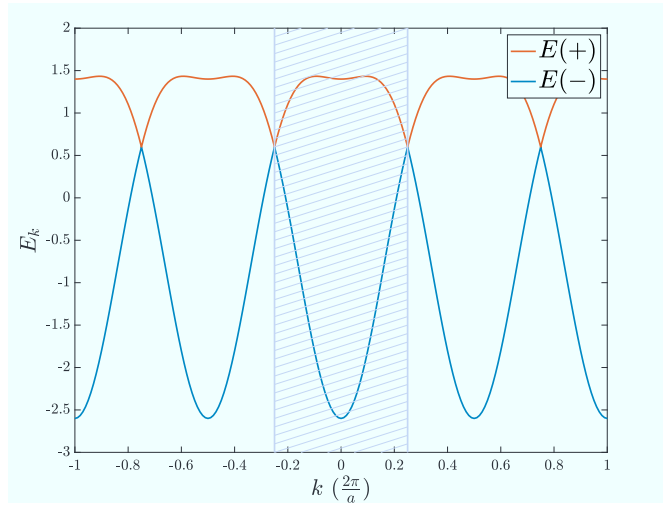


Figure 2.4.: Energy bands of a one-dimensional $\alpha\beta\alpha$ -chain with parameters $\varepsilon_\alpha = \varepsilon_\beta = 0$, $t = 1$, $t'_\alpha = t'_\beta = 0.3$. Dashed interval: the first Brillouin zone. Fig. 2.1 and Fig. 2.4 describe the same physics with two different perspectives.

2.5. The electronic structure of graphene

Graphene is a single two-dimensional layer of carbon atoms arranged in a hexagonal lattice.² Attributable to its many uncommon properties, graphene has been extensively studied for years. The material was isolated for the first time in 2004 by Andre Geim and Konstantin Novoselov at the University of Manchester [8]. Andre Geim and Konstantin Novoselov won the 2010 Nobel Prize in Physics “for groundbreaking experiments regarding

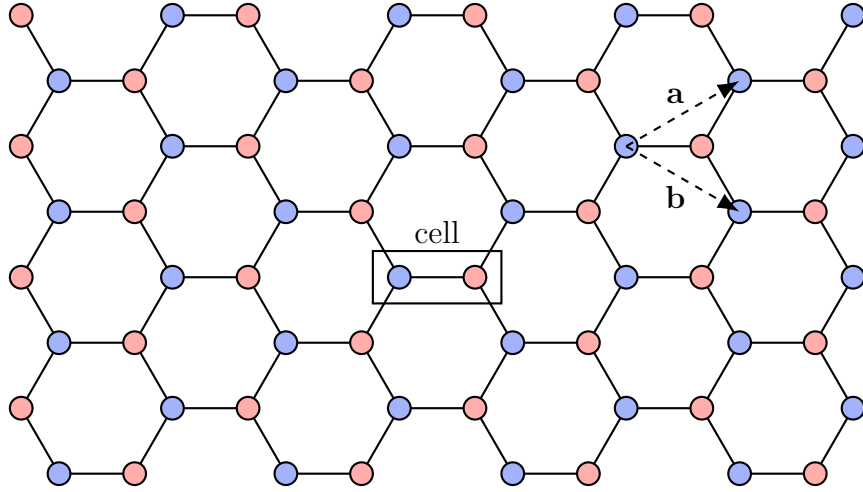
²Graphite, the most common allotrope of carbon, is basically a stack of graphene layers held together with weak bonds.

2. The tight-binding method

the two-dimensional material graphene”. Besides the novelties, it might be surprising that the electronic structures of graphene can be easily and successfully described by the tight-binding method.

On a graphene, each carbon atom is about ($a = 1.42 \text{ \AA} = 2.68 \text{ Bohr radius}$) apart from its three neighbors [9]. A carbon atom has the electronic structure $1s^2 2s^2 2p^2$. The two electrons in the $1s$ -shell have quite low energy. They can be considered as localized inside the atoms. Three electrons from $2s$, $2p_x$, $2p_y$ form the sp^2 hybrid orbitals. They contribute to the three 120° - σ -bonds. The remaining one electron is in the $2p_z$ orbital, which contributes to the π -bond. The π - π interactions among the half-filled $2p_z$ orbitals give the energy bands that are responsible for most of the electronic properties of graphene.

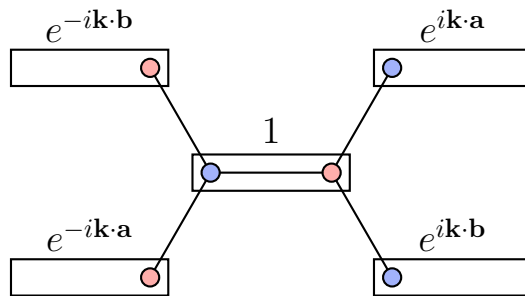
Now, we label half carbon atoms as α and half as β , and group them into cells. Why should we distinguish α and β although they are both carbon atoms? Because that is how the periodicity goes: If we translate α 's onto β 's, then the β 's will fall into the center of hexagons. The periodicity translates α 's to α 's.



This system has the lattice vectors:

$$\mathbf{A} = [\mathbf{a} \quad \mathbf{b}] = a \begin{bmatrix} \frac{3}{2} & \frac{3}{2} \\ \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad (2.32)$$

To simplify our problem, we restrict the electron to hop only to the nearest neighbors. Now we focus on one unit cell (with its nearest neighbours):



2.5. The electronic structure of graphene

Choosing the α -Bloch state and the β -Bloch state as our basis states:

$$c_{\mathbf{k}\alpha}^\dagger |0\rangle = \frac{1}{\sqrt{N}} \sum_j e^{i\mathbf{k}\cdot\mathbf{r}_j} c_{j\alpha}^\dagger |0\rangle \quad (2.33)$$

$$c_{\mathbf{k}\beta}^\dagger |0\rangle = \frac{1}{\sqrt{N}} \sum_j e^{i\mathbf{k}\cdot\mathbf{r}_j} c_{j\beta}^\dagger |0\rangle \quad (2.34)$$

The matrix representation of the Hamiltonian reads:

$$\mathbf{H} = \begin{bmatrix} \varepsilon & -t(1 + e^{-i\mathbf{k}\cdot\mathbf{a}} + e^{-i\mathbf{k}\cdot\mathbf{b}}) \\ -t(1 + e^{i\mathbf{k}\cdot\mathbf{a}} + e^{i\mathbf{k}\cdot\mathbf{b}}) & \varepsilon \end{bmatrix} \quad (2.35)$$

We obtain the two energy bands:

$$E_{\mathbf{k}} = \varepsilon \pm t\sqrt{3 + 2\cos(\mathbf{k}\cdot\mathbf{a}) + 2\cos(\mathbf{k}\cdot\mathbf{b}) + 2\cos(\mathbf{k}\cdot(\mathbf{a}-\mathbf{b}))} \quad (2.36)$$

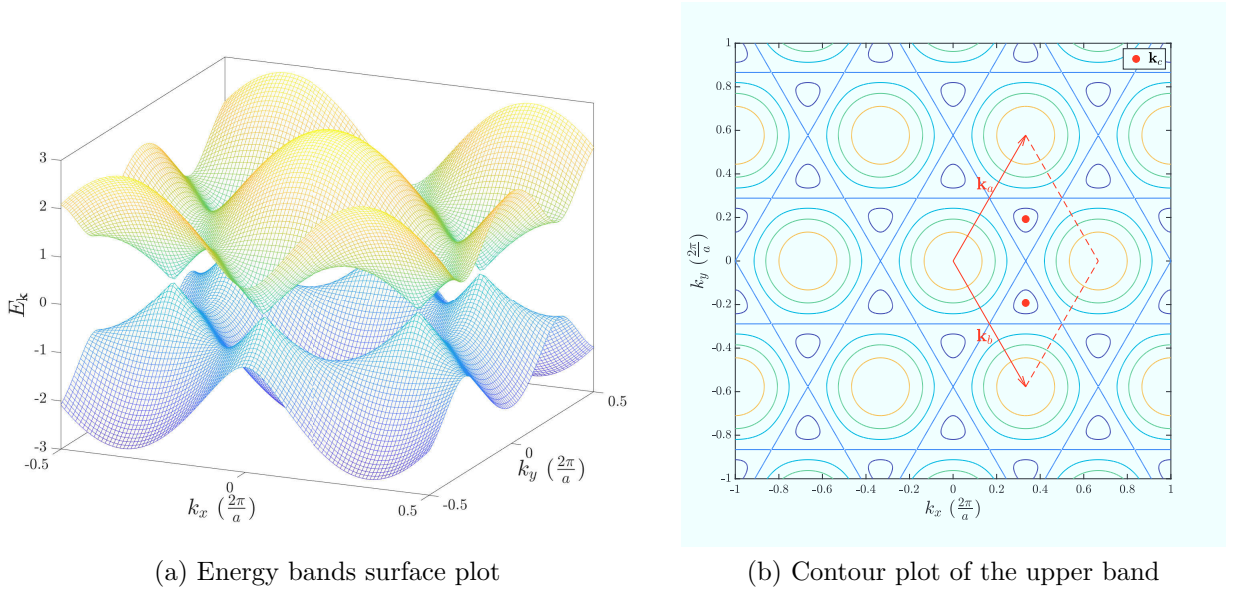


Figure 2.5.: (a) Energy bands of a two-dimensional graphene with parameters $\varepsilon = 0$ and $t = 1$. (b) The contour plot of the upper energy band surface. The area enclosed by the red parallelogram is a Brillouin zone. Two $\mathbf{k}_c$ points at the center of two Dirac cones are marked as red dots.

The two energy bands are plotted as surfaces on the (k_x, k_y) coordinate in Fig. 2.5a, and the upper band is plotted as contour in Fig. 2.5b. Because the system is half-filled (1 electron in $2p_z$), the upper and lower bands correspond to the conduction and valence bands, respectively. At the points where the two bands touch, they form the Dirac cones. Unlike insulators, there is no gap between the two bands; Unlike metals, there is no partially-filled band. Thus graphene is known as a zero-gap semiconductor (or semi-metal). The presence of the Dirac cones contributes to many impressive physical and electronic properties [9, 10].

2. The tight-binding method

Reciprocal lattice vectors:

Fig. 2.5b shows a contour plot of the upper energy band. The parallelogram formed by the reciprocal lattice vectors $\mathbf{k}_a$ and $\mathbf{k}_b$ encloses a Brillouin zone. $\mathbf{k}_a$ and $\mathbf{k}_b$ can be somehow *observed* from the contour plot. But how does one actually *calculate* them, for a given set of lattice vectors? The idea is the following:

If two $\mathbf{k}$ vectors are reproducing the same state, their dot product with each lattice site vector should be equal up to a multiple of 2π (recall Fig. 2.2).

$$\begin{aligned}\mathbf{k}_1 \cdot \mathbf{r}_i &= \mathbf{k}_2 \cdot \mathbf{r}_i + 2\pi n \\ \mathbf{k}_1 \cdot (\mathbf{A}\mathbf{i}) &= \mathbf{k}_2 \cdot (\mathbf{A}\mathbf{i}) + 2\pi n \\ (\mathbf{k}_2 - \mathbf{k}_1) \cdot (\mathbf{A}\mathbf{i}) &= 2\pi n \\ (\mathbf{G}\mathbf{j}) \cdot (\mathbf{A}\mathbf{i}) &= 2\pi n \\ \mathbf{j}^T \mathbf{G}^T \mathbf{A} \mathbf{i} &= 2\pi n\end{aligned}$$

Because $\mathbf{i}$ and $\mathbf{j}$ are arbitrary integer vectors, we have

$$\mathbf{G}^T \mathbf{A} = 2\pi \Rightarrow \boxed{\mathbf{G} = (2\pi \mathbf{A}^{-1})^T} \quad (2.37)$$

Therefore, for a given lattice vector set $\mathbf{A}$, we compute the reciprocal lattice vector set by taking inverse and transpose. Also note that the reciprocal lattice of a reciprocal lattice is the real space lattice itself: $(2\pi \mathbf{G}^{-1})^T = \mathbf{A}$.

The lattice vectors of graphene are:

$$\mathbf{A} = [\mathbf{a} \quad \mathbf{b}] = a \begin{bmatrix} \frac{3}{2} & \frac{3}{2} \\ \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad (2.38)$$

The corresponding reciprocal lattice vectors are, therefore,

$$\mathbf{G} = [\mathbf{k}_a \quad \mathbf{k}_b] = (2\pi \mathbf{A}^{-1})^T = \frac{2\pi}{a} \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \end{bmatrix} \quad (2.39)$$

which are the two vectors shown in Fig. 2.5b.

2.6. Carbon nanotube

Because of the zero-gap between the conduction and valence bands created by the π -bonds, graphene shows an efficient electrical conductivity. The conduction and valence bands “touch” at the $\mathbf{k}$ -points where the Dirac cones are formed. However, if boundary conditions are introduced and the $\mathbf{k}$ ’s are consequently quantized, the system could be either conducting or semiconducting, depending on the allowed $\mathbf{k}$ ’s. To be conducting, the allowed $\mathbf{k}$ ’s must include the Dirac cones. Let’s first check where the “cones” are.

Eqn. (2.36) tells us that the “touching points” are at the $\mathbf{k}_c$ points such that

$$3 + 2 \cos(\mathbf{k}_c \cdot \mathbf{a}) + 2 \cos(\mathbf{k}_c \cdot \mathbf{b}) + 2 \cos(\mathbf{k}_c \cdot (\mathbf{a} - \mathbf{b})) = 0 \quad (2.40)$$

To solve for $\mathbf{k}_c$, we first make an ansatz (hint from Fig. 2.5b):

$$\mathbf{k}_c = \alpha(\mathbf{k}_a - \mathbf{k}_b) \quad (2.41)$$

where $\mathbf{k}_a$ and $\mathbf{k}_b$ are the reciprocal lattice vectors. Hence, (use the orthogonal relation between real lattice vectors and reciprocal lattice vectors)

$$3 + 2 \cos(2\pi\alpha) + 2 \cos(-2\pi\alpha) + 2 \cos(4\pi\alpha) = 0$$

$$3 + 4 \cos(2\pi\alpha) + 2 \cos(4\pi\alpha) = 0$$

$$4 \cos^2(2\pi\alpha) + 4 \cos(2\pi\alpha) + 1 = 0$$

$$\cos(2\pi\alpha) = -\frac{1}{2} \Rightarrow \alpha = \pm \frac{1}{3} \quad (2.42)$$

Therefore,

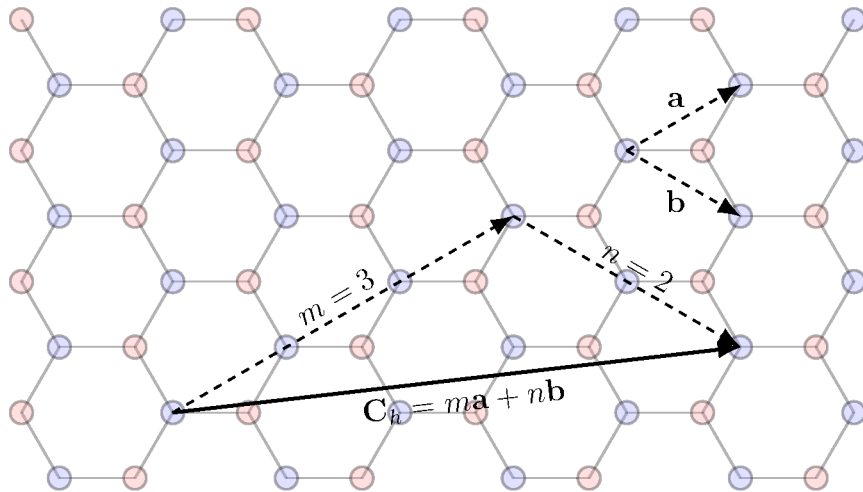
$$\mathbf{k}_c = \pm \frac{1}{3}(\mathbf{k}_a - \mathbf{k}_b) \quad (2.43)$$

And we can translate them in integer multiples of $\mathbf{k}_a$ and $\mathbf{k}_b$: ($i, j \in \mathbb{Z}$)

$$\mathbf{k}_c = i\mathbf{k}_a + j\mathbf{k}_b \pm \frac{1}{3}(\mathbf{k}_a - \mathbf{k}_b) \quad (2.44)$$

Now we would like to introduce some boundary condition. We fold the graphene sheet around to make a carbon nanotube. This “roll-up” follows one direction, called the chiral vector:

$$\mathbf{C}_h = m\mathbf{a} + n\mathbf{b} \quad (2.45)$$



Here we need some visual-spatial imagination: (can you imagine?)

2. The tight-binding method

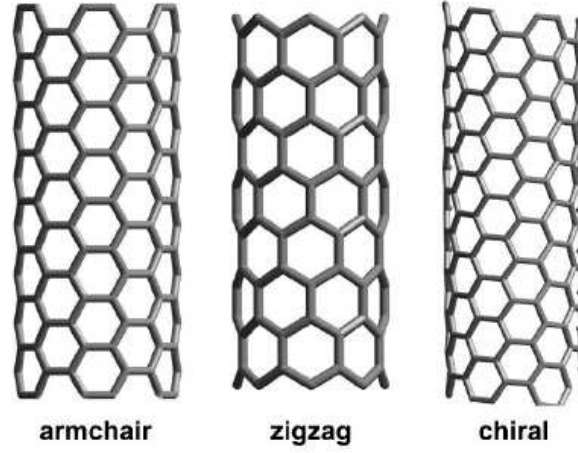


Figure 2.6.: Structure of carbon nanotubes depending on the roll-up direction. Image from: <https://coecs.ou.edu/Brian.P.Grady/nanotube.html>.

- If $\mathbf{C}_h = m\mathbf{a} + n\mathbf{b}$, the roll-up gives armchair-like ends;
- If $\mathbf{C}_h = m\mathbf{a}$ or $\mathbf{C}_h = n\mathbf{b}$, the roll-up gives zig-zag ends;
- Otherwise, the ends are a bit tilted.

Once the graphene is rolled up, we have a well defined periodic boundary condition: ($l \in \mathbb{Z}$)

$$\mathbf{k} \cdot \mathbf{C}_h = 2\pi l \quad (2.46)$$

This defines the allowed $\mathbf{k}$'s. If the allowed $\mathbf{k}$'s contain $\mathbf{k}_c$ (2.44), then $\mathbf{k}_c$ must fulfill (2.46): ($i, j, l, m, n \in \mathbb{Z}$)

$$\begin{aligned} \mathbf{k}_c \cdot \mathbf{C}_h &= 2\pi l \\ \left[(i \pm \frac{1}{3})\mathbf{k}_a + (j \mp \frac{1}{3})\mathbf{k}_b \right] \cdot [m\mathbf{a} + n\mathbf{b}] &= 2\pi l \\ im \pm \frac{1}{3}m + jn \mp \frac{1}{3}n &= l \\ \pm \frac{1}{3}(m - n) &= l \\ m - n &= 3l \end{aligned} \quad (2.47)$$

The simple relation (2.47) leads to the following statement:

- The conductance of a carbon nanotube depends on the chiral vector $\mathbf{C}_h = m\mathbf{a} + n\mathbf{b}$.
 - If $m - n$ is an integer multiple of 3, the tube is conducting (the conduction and valence bands touch);
 - Otherwise, the tube is semiconducting (the conduction and valence bands have a gap);

Some quick facts: armchair tubes are always metallic because $m = n$; zig-zag tubes are sometimes metallic, depending on whether or not m or n is an integer multiple of 3.

In Fig. 2.7, we visualize the allowed $\mathbf{k}$'s in one Brillouin zone, for different cases of (m, n) . Notice that due to the boundary condition, the $\mathbf{k}$ -surface is discretized to $\mathbf{k}$ -lines. One can verify that for a given (m, n) , whether or not the $\mathbf{k}$ -lines include the $\mathbf{k}_c$'s at the Dirac cones.

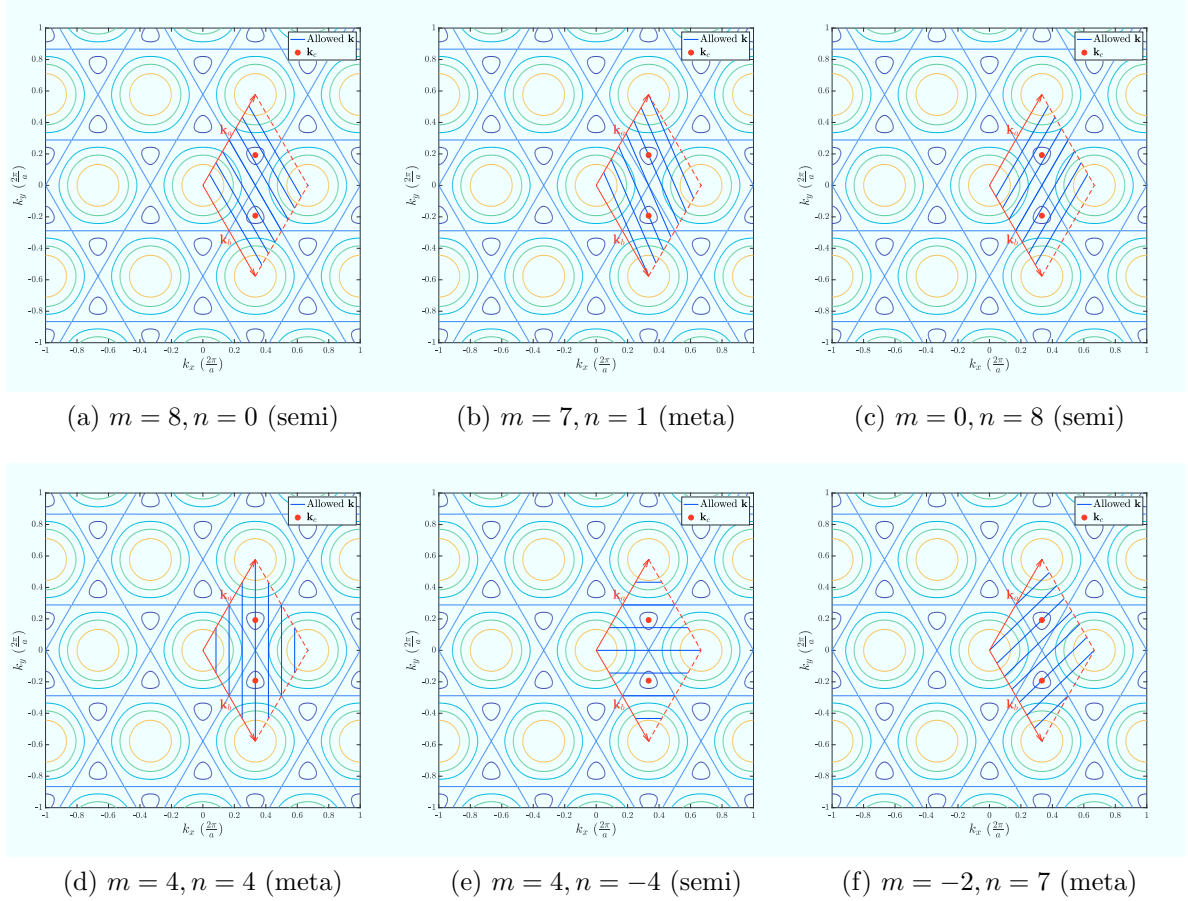


Figure 2.7.: Allowed $\mathbf{k}$'s in one Brillouin zone. If the allowed $\mathbf{k}$'s (blue lines) go cross the Dirac cones (red dots), the system is metallic (meta); otherwise semiconducting (semi).

3. Self-consistent field solver for atomic systems

3.1. The starting point

In the previous chapter, we discussed the tight-binding method with some testing numerical values for the matrix elements to demonstrate the concepts. Certainly, for constructing realistic model Hamiltonians, we need realistic basis functions and the corresponding matrix elements. The starting point of this thesis is density functional calculations for individual atoms and ions to obtain realistic basis orbitals.

The purpose of this chapter is to develop and document a self-consistent field (SCF) solver that solves the atomic systems in the framework of density functional theory (DFT) [11, 12]. The main work during the development is to build an efficient and robust eigen-state solver that solves for the atomic orbitals. We use techniques such as Numerov's method and the first-order perturbation theory to achieve high-order numerical accuracy and fast convergence rate for the eigen-state solutions. Our aim is to build an elegant and compact code for the SCF calculations. By the end of this chapter, we should have a clear and well explained working Python code.¹

3.2. Hydrogen-like system

To start with, we solve the one-electron or hydrogen-like system numerically, for which we can compare the results with the analytical solutions. After a reliable solver is developed, we extend the program to solve a general atomic system in the Kohn-Sham [12] approach. Although the solutions for hydrogen-like systems are known analytically (see Appendix A), solving the problem numerically is not a trivial task. The main points are:

1. How to properly discretize the grid in space;
2. How to properly integrate the differential equation;
3. How to obtain the eigen-energies efficiently and accurately.

¹By default, we use Python version 3.

3. Self-consistent field solver for atomic systems

For a hydrogen-like system, the Schrödinger equation (in a.u.) reads,

$$\left[-\frac{1}{2}\nabla^2 + V(r) \right] \varphi = E\varphi \quad (3.1)$$

where $V(r) = -Z/r$ is the (spherically symmetric) nuclear potential. By separation of variables $\varphi(r, \theta, \phi) = R(r)Y(\theta, \phi)$, Eqn. (3.1) splits into two equations, namely,

$$\text{Radial equation:} \quad \frac{d}{dr} \left(r^2 \frac{dR}{dr} \right) - 2r^2 [V(r) - E] R = l(l+1)R \quad (3.2)$$

$$\text{Angular equation:} \quad \frac{1}{\sin\theta} \frac{\partial}{\partial\theta} \left(\sin\theta \frac{\partial Y}{\partial\theta} \right) + \frac{1}{\sin^2\theta} \frac{\partial^2 Y}{\partial\phi^2} = -l(l+1)Y \quad (3.3)$$

Solutions of the angular equation (3.3) are the well known spherical harmonics. Our task is to solve (3.2) numerically, as later we will replace $V(r)$ by an effective mean-field potential $V_{\text{KS}}(r)$. Here we introduce a change of variables $u(r) \equiv rR(r)$. As a result, the radial equation (3.2) simplifies dramatically to the form of a one-dimensional Schrödinger equation:

$$\left[-\frac{1}{2} \frac{d^2}{dr^2} + V(r) + \frac{l(l+1)}{2r^2} \right] u = Eu \quad (3.4)$$

with asymptotic behaviors: (this helps us to set up the initial integration values)

$$u(r) \propto r^{l+1} \quad \text{as } r \rightarrow 0 \quad (3.5)$$

$$u(r) \propto e^{-\kappa r}, \quad \kappa = \sqrt{-2E} \quad \text{as } r \rightarrow \infty \quad (3.6)$$

Our task is to solve (3.4) for the radial wave functions $u(r)$ and the allowed energies E .

Hydrogen-like solutions:

The hydrogen and hydrogen-like solutions are related by a simple “rescaling”.

For a hydrogen atom, Eqn. (3.4) reads,

$$\left[-\frac{1}{2} \frac{d^2}{dr^2} - \frac{1}{r} + \frac{l(l+1)}{2r^2} \right] u_{\text{H}}(r) = E_{\text{H}} u_{\text{H}}(r) \quad (3.7)$$

For a hydrogen-like atom, Eqn. (3.4) reads,

$$\left[-\frac{1}{2} \frac{d^2}{dr^2} - \frac{Z}{r} + \frac{l(l+1)}{2r^2} \right] u_{\text{Z}}(r) = E_{\text{Z}} u_{\text{Z}}(r) \quad (3.8)$$

Now, if we rescale $\rho = Zr$ and $E_{\text{Z}} = Z^2 E_{\text{H}}$, Eqn. (3.8) becomes,

$$\left[-\frac{1}{2} \frac{d^2}{d\rho^2} - \frac{1}{\rho} + \frac{l(l+1)}{2\rho^2} \right] u_{\text{H}}(\rho) = E_{\text{H}} u_{\text{H}}(\rho) \quad (3.9)$$

which is identical to the hydrogen system. Solving (3.9) yields the same eigenfunctions and eigen-values as in solving (3.7). Attention: $u_{\text{H}}(\rho)$ solves (3.9), it is,

however, not properly normalized:

$$\int_0^\infty dr |u_H(\rho)|^2 = \frac{1}{Z} \int_0^\infty d\rho |u_H(\rho)|^2 = \frac{1}{Z} \quad (3.10)$$

Therefore, the hydrogen-like solutions are related to the hydrogen solutions by

$$\boxed{u_Z(r) = Z^{\frac{1}{2}} u_H(Zr)} \quad \text{and} \quad \boxed{E_Z = Z^2 E_H} \quad (3.11)$$

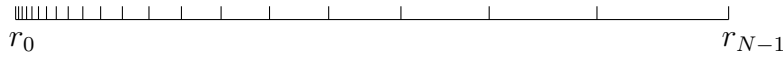
See Table A.1 in Appendix A for the explicit formulas.

3.3. Logarithmic grid

Atomic wave functions tend to oscillate stronger around the nucleus. To resolve the sharp features, our choice is to use a logarithmic grid [13, 14, 15]. A logarithmic grid is certainly not the unique choice. One can use any adaptive grid as long as it can properly represent the wave function [13]. The advantage of the logarithmic grid is that it uses a change of variable technique and the original ODE can be transformed smoothly into an equivalent ODE problem on a uniform grid.

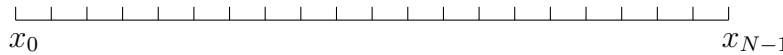
Define the logarithmic grid: $0 < r_0 < r_1 < \dots < r_{N-1} < \infty$, where

$$r_i = \frac{1}{Z} e^{x_i} \quad (3.12)$$



and x is a uniformly distributed grid

$$x_i = x_0 + i\Delta x \quad (3.13)$$



The factor $\frac{1}{Z}$ in (3.12) is used due to the scaling behavior of hydrogen-like solutions. Notice that we cannot take $r_0 = 0$, as $x_0 = \ln(Zr_0)$ would go $-\infty$. To set up a grid, we provide four arguments: the atomic number Z ; the minimum $\mathbf{rmin}$; the maximum $\mathbf{rmax}$; and the spacing on the uniform grid $\mathbf{dx}$ (we do not input $\mathbf{dr}$, since it is not a constant).

We create a file `grid.py`, which sets up the `Grid` object (Algorithm 3.1).

Algorithm 3.1: The `Grid` class (`grid.py`).

```

1 import math
2 import numpy as np
3
4 class Grid:
5     def __init__(self, Z, rmin, rmax, dx):
```

3. Self-consistent field solver for atomic systems

```

6      self.Z = Z                # Atomic number
7      xmin = math.log(Z*rmin)   # Minimum of x
8      xmax = math.log(Z*rmax)   # Maximum of x
9      self.dx = dx              # Delta x
10     self.x = np.arange(xmin, xmax, dx) # Uniform grid
11     self.r = np.exp(self.x)/Z  # Logarithmic grid
12     self.V = -Z/self.r         # Potential on the grid
13     self.N = len(self.r)       # Number of grid points

```

For a given atomic number Z (e.g. $Z = 1$ for hydrogen), we set up the numerical grid:

$$\{r_{\min} = 10^{-6}/Z; \quad r_{\max} = 50; \quad \Delta x = 0.005\}$$

The choice of the minimum and the maximum of the radial grid $r_{\min} = 10^{-6}/Z$ and $r_{\max} = 50$ agrees with Reference [13]. The minimum of the radial grid depends on the atomic number Z , because the higher the nuclear charge is, the closer the wave function will be attracted to the origin. On the other hand, the maximum of the radial grid is fixed. This consideration is from the fact that the size of each atom will be roughly the same after the self-consistent calculations. The uniform grid resolution Δx is chosen to be 0.005. Further increasing of the grid resolution wouldn't affect the calculations significantly. As a reference, for the chosen grid set up, the number of grid points $\#$ are:

$$Z = 1 \rightarrow \# = 3546; \quad Z = 10 \rightarrow \# = 4007; \quad Z = 100 \rightarrow \# = 4467$$

The original problem u on the radial coordinate r can be transformed to a rescaled problem² $\tilde{u} \equiv u/\sqrt{r}$ on the uniform coordinate x . The idea is to replace the second derivative d^2u/dr^2 in terms of $d^2\tilde{u}/dx^2$:

$$\frac{d^2u}{dr^2} = -\frac{1}{4}r^{-3/2}\tilde{u} + r^{-3/2}\frac{d^2\tilde{u}}{dx^2} \quad (3.14)$$

Substituting (3.14) into (3.4), one obtains

$$\left[-\frac{1}{2}\frac{d^2}{dx^2} + r^2V(r) + \frac{1}{2}\left(l + \frac{1}{2}\right)^2 \right] \tilde{u} = r^2E\tilde{u} \quad (3.15)$$

which is the transformed problem. Now, instead of solving the original ODE in (3.4) on a logarithmic grid, we solve this transformed ODE in (3.15) on the uniform grid. The advantage is that we can easily discretize the second derivative $d^2\tilde{u}/dx^2$, since the grid x_i is uniformly spaced. For convenience, we rewrite (3.15) in a more compact way:

$$\frac{d^2\tilde{u}}{dx^2} = -k_i^2\tilde{u}_i \quad (3.16)$$

where,

$$k_i^2 \equiv 2r_i^2E - 2r_i^2V(r_i) - \left(l + \frac{1}{2}\right)^2 \quad (3.17)$$

²If there weren't this rescaling, the transformed problem would involve a first-order derivative du/dx , which would not be suitable for later applying Numerov's method.

3.4. Numerov's method

To discretize $d^2\tilde{u}/dx^2$, the simplest way is perhaps in the finite-difference form:

$$\frac{d^2\tilde{u}}{dx^2} \approx \frac{\tilde{u}_{i+1} - 2\tilde{u}_i + \tilde{u}_{i-1}}{\Delta x^2} \quad (3.18)$$

which is numerically accurate up to $\mathcal{O}(\Delta x^2)$. If we further expand (3.18), we get higher order terms:

$$\frac{d^2\tilde{u}}{dx^2} = \frac{\tilde{u}_{i+1} - 2\tilde{u}_i + \tilde{u}_{i-1}}{\Delta x^2} - \frac{1}{12}\tilde{u}_i^{(4)}\Delta x^2 + \mathcal{O}(\Delta x^4) \quad (3.19)$$

But the fourth derivative $\tilde{u}_i^{(4)}$ can be also written in the finite-difference form:

$$\frac{d^2\tilde{u}}{dx^2} = \frac{\tilde{u}_{i+1} - 2\tilde{u}_i + \tilde{u}_{i-1}}{\Delta x^2} - \frac{1}{12}\frac{\tilde{u}_{i+1}'' - 2\tilde{u}_i'' + \tilde{u}_{i-1}''}{\Delta x^2}\Delta x^2 + \mathcal{O}(\Delta x^4) \quad (3.20)$$

Here comes the Numerov trick, instead of treating the second derivative $\tilde{u}_i''$ in (3.20) numerically, one can simply replace the $\tilde{u}_i''$ by the relation from the original ODE in (3.16). Thus, the second derivative discretizes to

$$\frac{d^2\tilde{u}}{dx^2} \approx \frac{\tilde{u}_{i+1} - 2\tilde{u}_i + \tilde{u}_{i-1}}{\Delta x^2} + \frac{1}{12}(k_{i+1}^2\tilde{u}_{i+1} - 2k_i^2\tilde{u}_i + k_{i-1}^2\tilde{u}_{i-1}) \quad (3.21)$$

which is numerically accurate up to $\mathcal{O}(\Delta x^4)$. Comparing (3.21) with (3.18), we see that with the same amount of computational effort, the numerical scheme improves from 2nd-order to 4th-order accuracy “for free”. That is the power of Numerov's method. It works only because the ODE in (3.16) does not involve a first order derivative.

Now we substitute (3.21) into (3.16), we obtain the following relation:

$$\boxed{\tilde{u}_{i\pm 1} = \frac{(2 - \frac{5\Delta x^2}{6}k_i^2)\tilde{u}_i - (1 + \frac{\Delta x^2}{12}k_{i\mp 1}^2)\tilde{u}_{i\mp 1}}{1 + \frac{\Delta x^2}{12}k_{i\pm 1}^2}} \quad (3.22)$$

which is perhaps the most important formula that will enter into the code. This simple 3-point recursion states that: with the knowledge of $\tilde{u}_{i-1}$ and $\tilde{u}_i$, we can compute $\tilde{u}_{i+1}$. Or from the other direction, compute $\tilde{u}_{i-1}$ from $\tilde{u}_i$ and $\tilde{u}_{i+1}$.

3.5. The shooting and matching methods

We initialize the wave function values according to the asymptotic behaviors (3.5) and (3.6), shown in Table 3.1. One could perform Numerov's integration (3.22) either in the forward direction starting from initial values $\tilde{u}_0$ and $\tilde{u}_1$, or in the backward direction starting from initial values $\tilde{u}_{N-1}$ and $\tilde{u}_{N-2}$. To ensure numerical stability, we perform integrations from both the forward and backward directions. On the radial grid, the forward and backward results will meet at a point, which we call the “matching point”.

3. Self-consistent field solver for atomic systems

Table 3.1.: Initializations for the forward and the backward wave functions. ($\kappa = \sqrt{-2E}$)

	Initialization
Forward	$\tilde{u}_0 = r_0^{l+1}/\sqrt{r_0}$ $\tilde{u}_1 = r_1^{l+1}/\sqrt{r_1}$
Backward	$\tilde{u}_{N-1} = e^{-\kappa r_{N-1}}/\sqrt{r_{N-1}}$ $\tilde{u}_{N-2} = e^{-\kappa r_{N-2}}/\sqrt{r_{N-2}}$

Because of numerical instability, the wave functions will diverge out quickly if integrating into classically forbidden regions. Our matching point is therefore chosen to be around the classical turning point. More specifically, we choose the matching point r_M to be the first grid point (forward direction) which fulfills the condition $E \leq V(r_i)$. The choice of r_M is not very critical: we simply compare the energy E with the monotonic $V(r)$ which has a unique root at $E = V(r)$, while the “true” classical turning point should be at $E = V_{\text{eff}}(r) = V(r) + \frac{l(l+1)}{2r^2}$ which can have no root, one root, or two roots due to the centrifugal term, that complicates the problem and is not really necessary. At the matching point we have two values $\tilde{u}_{\text{FM}}$ and $\tilde{u}_{\text{BM}}$ computed from the forward and backward Numerov’s integrations, respectively. Naturally, $\tilde{u}_{\text{FM}}$ and $\tilde{u}_{\text{BM}}$ should be equal, as they represent the same function value. But the initial conditions (Table 3.1) are assigned up to a scaling, which means a rescaling should be applied to either the forward or the backward part to match the two pieces. Finally, the resulting wave function should be normalized. Notice that because of the change of variable (3.12), we have $dr = rdx$. Additionally, we are working with the rescaled function $\tilde{u} \equiv u/\sqrt{r}$. Thus, the normalization of “ $\tilde{u}$ ” becomes:

$$\tilde{u} \leftarrow \tilde{u} / \sqrt{\int_{-\infty}^{+\infty} dx r^2 |\tilde{u}|^2} \quad (3.23)$$

We create a file `orbit.py`, which contains the `Orbit` class with its relevant methods (Algorithm 3.2). At this stage, we provide the `shoot` method, which takes an arbitrary energy and produces the corresponding wave function. The method is called “shoot” because it is the basic function that tries to find an eigen-energy by trial-and-error (like shooting a basket ball into a basket by trying different throws), which we will explain soon. There are a couple of technical details that are implemented in the code: If E is above the maximum value of V (which is an unusual case since we are looking for bound states), no matching point will exist. In this case, we simply take $M=N/2$, and let the forward and backward functions meet in the middle of the grid. Moreover, since it is not in a bound state, we artificially set an infinite potential well at r_{N-1} , such that one can still look for the eigen-energy. Another pitfall is in the backward initialization: the numerical values of $\tilde{u}_{N-1}$ and $\tilde{u}_{N-2}$ can be extremely small due to the exponential decaying behavior. Numerically, they can easily go underflow and become zeros (or too small to have enough significant digits). This issue has to be taken into account in the program as well.

Algorithm 3.2: A preliminary Orbit class (orbit.py).

```

1 import math
2 import numpy as np
3 import scipy.integrate as sp
4
5 class Orbit:
6     def __init__(self, G, n, l):
7         self.G = G          # Grid
8         self.n = n          # Principal quantum number
9         self.l = l          # Angular momentum quantum number
10        self.R = np.zeros(G.N) # Radial wave function
11        self.u = np.zeros(G.N) # u = r*R
12        self.E = None        # Orbital energy
13        self.M = None        # Matching point index
14
15    # Shoot a wave function with given E
16    def shoot(self, E):
17        # Collect attributes (overwrite u to re-use the memory space)
18        (N, dx, r, V, l, u) = (self.G.N, self.G.dx, self.G.r, self.G.V, self.l, self.u)
19
20        # Matching point
21        M = np.where(E<V)[0][0] if E<V[-2] else N//2
22
23        # During the calculation, u is the rescaled u/sqrt(r)
24        # Forward initial condition [0][1]-->
25        u[:2] = r[:2]**(1+0.5)
26        # Backward initial condition <--[end-1][end]
27        end = N-1
28        if E<V[-2]:
29            # Exponential decay, avoid numerical underflow
30            tail = np.exp(-math.sqrt(abs(2*E))*r)/np.sqrt(r)
31            end = np.where(tail>1e-16)[0][-1] # After [end], tail too small
32            u[end-1:] = tail[end-1:]
33        else:
34            # Energy too high, use hard boundary condition
35            u[-2:] = [1e-8, 0.0]
36
37        # Numerov method
38        kk = 2*(r**2*(E-V) - 0.5*(1+0.5)**2)
39        A = 2 - dx*dx*5/6*kk
40        B = 1 + dx*dx/12*kk
41        # Forward integration [0][1]-->[M]
42        for i in range(2, M+1):
43            u[i] = (A[i-1]*u[i-1] - B[i-2]*u[i-2]) / B[i]
44        FM = u[M]
45        # Backward integration [M]<--[end-1][end]
46        for i in range(end-2, M-1, -1):
47            u[i] = (A[i+1]*u[i+1] - B[i+2]*u[i+2]) / B[i]
48        BM = u[M]
49        # Connect forward and backward parts
50        u[M:] *= FM/BM
51
52        # Normalization
53        u /= math.sqrt(sp.simps(r**2*u**2, dx=dx))
54
55        # Collect results and rescale u back to u = r*R
56        (self.R, self.u, self.E, self.M) = (u/np.sqrt(r), u*np.sqrt(r), E, M)

```

For loops in Python:

It is perhaps a common knowledge that when one writes a code in Python, one should avoid intensive for loops. For instance, the piece of normalization code is written with

3. Self-consistent field solver for atomic systems

`simps` (Simpson's rule) from the `scipy` package, or simply use (but less accurate) the `trapz` (Trapezoidal rule) from the `numpy` package, instead of an explicit for loop which will reduce both the performance and elegance of the code. Unfortunately, the piece of Numerov's iteration has to stay in an explicit loop (at least up to my current knowledge). The reason is that during the iteration, each step has a dependence to the results from previous steps, unlike simple operations such as array summations or mapping to functions, etc..

We are now in a position to run a test. The `run.py` file is shown in Algorithm 3.3. At this point, setting up the principal quantum number n has no significance. It does not play any role in the `shoot` method. Later, we will use it to compare with the number of nodes in the wave function to check if the wave function is the one that corresponds to the quantum numbers n and l . In the test, we set up a hydrogen system ($Z = 1$) with a $1s$ orbital. We test three orbital energies $E = -0.6, -0.5, -0.4$ (Hartree), and the resulting wave functions are plotted in Fig. 3.1. On top of the plots, we put the exact hydrogen $1s$ orbital ($u = 2re^{-r}$) in dashed lines as a reference. Analytically, we know that the $1s$ orbital has an eigen-energy $-1/2$ Hartree. Numerically, we see that when the testing energies differ from the eigen-energy, the resulting wave functions will have a kink at the matching point. The forward and backward wave functions match smoothly when the testing energy agrees with the analytical value.

Algorithm 3.3: A test run.

```
1 from grid import Grid
2 from orbit import Orbit
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 # Set up grid
7 G = Grid(1, 1e-6, 50, 0.005)
8
9 # Set up orbit
10 orb = Orbit(G, n=1, l=0)
11
12 for E in [-0.6, -0.5, -0.4]:
13     # Shoot with a guess E
14     orb.shoot(E)
15
16     # Plot
17     plt.plot(G.r[:orb.M], orb.u[:orb.M], '-b', label='Forward')
18     plt.plot(G.r[orb.M:], orb.u[orb.M:], '-r', label='Backward')
19     plt.plot(G.r, 2*G.r*np.exp(-G.r), '--m', label='Exact')
20     plt.legend(loc='best')
21     plt.show()
```

3.6. Predict ΔE by first-order perturbation theory

Suppose we didn't know the analytical solution of hydrogen, and we are solving for the hydrogen $1s$ orbital. We make a guess $E = -0.6$ and get the result shown in Fig. 3.1a.

3.6. Predict ΔE by first-order perturbation theory

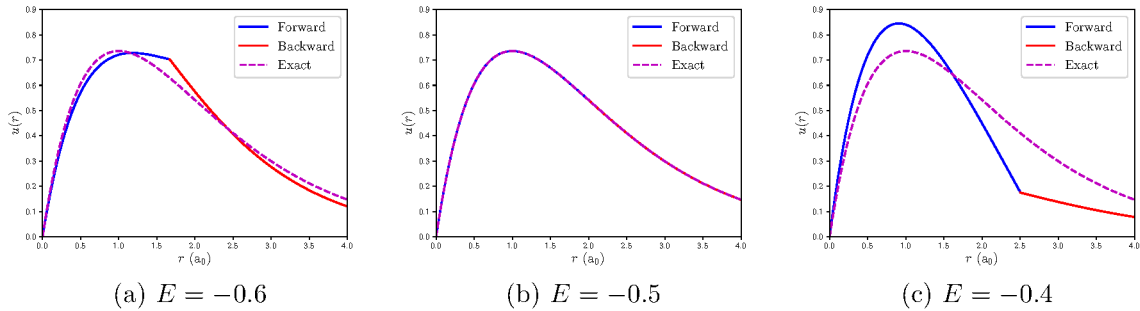


Figure 3.1.: Three tests of the **shoot** method with orbital energies $E = -0.6, -0.5, -0.4$ (Hartree). When the testing energies differ from the eigen-energy, the resulting wave functions will have a kink at the matching point. The forward and backward wave functions match smoothly when the testing energy agrees with the analytical value.

We see that there is kink. Now, should we increase the energy a bit, or reduce, and by how much? How do we make a better guess towards the true eigen-energy?

A simple strategy is to use a “bisection” method that repeatedly halves an energy interval until a root is reached up to a tolerance. The bisection method is easy to implement and robust, but it is relatively slow. Here we introduce a new method based on the first-order perturbation theory which speeds up the root-finding process dramatically.

The idea is the following:

1. We make a guess $\hat{E}$, which produces a kinked wave function $\hat{u}$ (to be consistent with the formalism in (3.22), we shall use the rescaled $\tilde{u} \equiv \hat{u}/\sqrt{r}$ in the calculation);
2. $\hat{u}$ is not an eigen-function of the potential V ;
3. However, since we are working on a discrete grid, the numerical array that represents $\hat{u}$, must be an “eigen-vector” of some unknown potential $\hat{V}$;
4. V and $\hat{V}$ are related by a perturbation ΔV :

$$V = \hat{V} + \Delta V \quad (3.24)$$

5. When $\hat{E}$ is close enough to the true eigen-energy E , according to the first-order perturbation theory, we can approximate the eigen-energy by

$$E \approx \hat{E} + \Delta E \quad (3.25)$$

where,

$$\Delta E = \langle \hat{u} | \Delta V | \hat{u} \rangle = \int_0^\infty dr |\hat{u}|^2 \Delta V = \int_{-\infty}^{+\infty} dx r^2 |\tilde{u}|^2 \Delta V \quad (3.26)$$

If we can work out the perturbation potential ΔV , then we can compute the energy correction ΔE . In practice, we produce an approximated ΔV as described below:

3. Self-consistent field solver for atomic systems

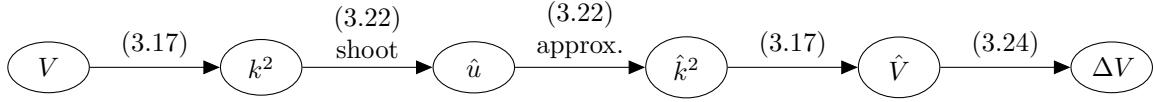


Figure 3.2.: Estimate the perturbed potential.

For a given potential V and a guessed energy, we obtain k^2 from Eqn. (3.17). Using the Numerov method (3.22), we obtain the wave function (presumably a kinked one) $\hat{u}$ from the shoot and match. Now, we go a step reverse: we produce $\hat{k}^2$ from $\hat{u}$, assuming that $\hat{k}_i^2 = k_i^2$ everywhere except at the matching point M . This assumption implies that the perturbation ΔV is everywhere zero except at the matching point:

$$\Delta V_i = \Delta V_M \delta_{iM} \quad (3.27)$$

In fact, a delta potential produces a kinked wave function (see Griffiths [16] on the delta-function potential). At the matching point, $\hat{k}_M^2$ differs from k_M^2 , and its value can be computed from the recursion relation (3.22):

$$\hat{k}_M^2 = \frac{6}{5\Delta x^2} \left[2 - \left(1 + \frac{\Delta x^2}{12} k_{M+1}^2 \right) \frac{\tilde{u}_{M+1}}{\tilde{u}_M} - \left(1 + \frac{\Delta x^2}{12} k_{M-1}^2 \right) \frac{\tilde{u}_{M-1}}{\tilde{u}_M} \right] \quad (3.28)$$

From (3.17) and (3.24), we obtain

$$\Delta V_M = V_M - \hat{V}_M = \frac{\hat{k}_M^2 - k_M^2}{2r_M^2} \quad (3.29)$$

Substitute (3.29) back to (3.26), we conclude, (notice that the domain of the integral reduces to the small interval at the matching point)

$$\boxed{\Delta E = \Delta x r_M^2 |\tilde{u}_M|^2 \Delta V_M} \quad (3.30)$$

ΔE becomes zeros when ΔV_M is zero. The energy correction ΔE is also an indicator of the smoothness of the wave function. If $\Delta E \approx 0$, then we know the resulting wave function is kink-free, which means we have converged to an eigen-state (see Figs. 3.3b and 3.3h). Normally, we do not expect $u_M = 0$, as the matching point is at the classical turning point, where no node should occur. However, when the number of nodes changes, u_M must change a sign (see Fig. 3.3d), which means $u_M = 0$ does occur at a very critical energy which is not an eigen-energy. To better explain this matter, I reproduce the hydrogen $l = 0$ system (was shown in Fig. 3.1) for 9 testing energies and print the corresponding proposed ΔE 's in Table 3.2. We see that, E_{1s} and E_{2s} are eigen-energies ($\Delta E = 0$), which are (up to numerical errors) equal to -0.5 and -0.125 , respectively. The neighboring energies produce ΔE 's that move towards the eigen-energies, thus E_{1s} and E_{2s} are attractive fixed points. However, E_x is not an eigen-energy although $\Delta E = 0$. The neighboring energies produce ΔE 's that move away from E_x . In practice, we do not worry about the situations for E_x as they are unstable points that the iterative solver will not converge to.

3.6. Predict ΔE by first-order perturbation theory

Table 3.2.: Testing energy E versus proposed ΔE . E_{1s} and E_{2s} are eigen-energies ($\Delta E = 0$), which are (up to numerical errors) equal to -0.5 and -0.125 , respectively. The neighboring energies produce ΔE 's that move towards the eigen-energies, thus E_{1s} and E_{2s} are attractive fixed points. However, E_x is not an eigen-energy although $\Delta E = 0$. The neighboring energies produce ΔE 's that move away from E_x .

E	ΔE	Direction
-0.115	-0.006535	$\downarrow$
-0.125	-3×10^{-12}	$\downarrow$
E_{2s}	0	attractive
-0.15	$+0.029706$	$\uparrow$
-0.3	$+0.218502$	$\uparrow$
-0.35	$+0.078240$	$\uparrow$
E_x	0	unstable
-0.375	-0.000144	$\downarrow$
-0.4	-0.040794	$\downarrow$
-0.5	-6×10^{-13}	$\downarrow$
E_{1s}	0	attractive
-0.6	$+0.111559$	$\uparrow$

3. Self-consistent field solver for atomic systems

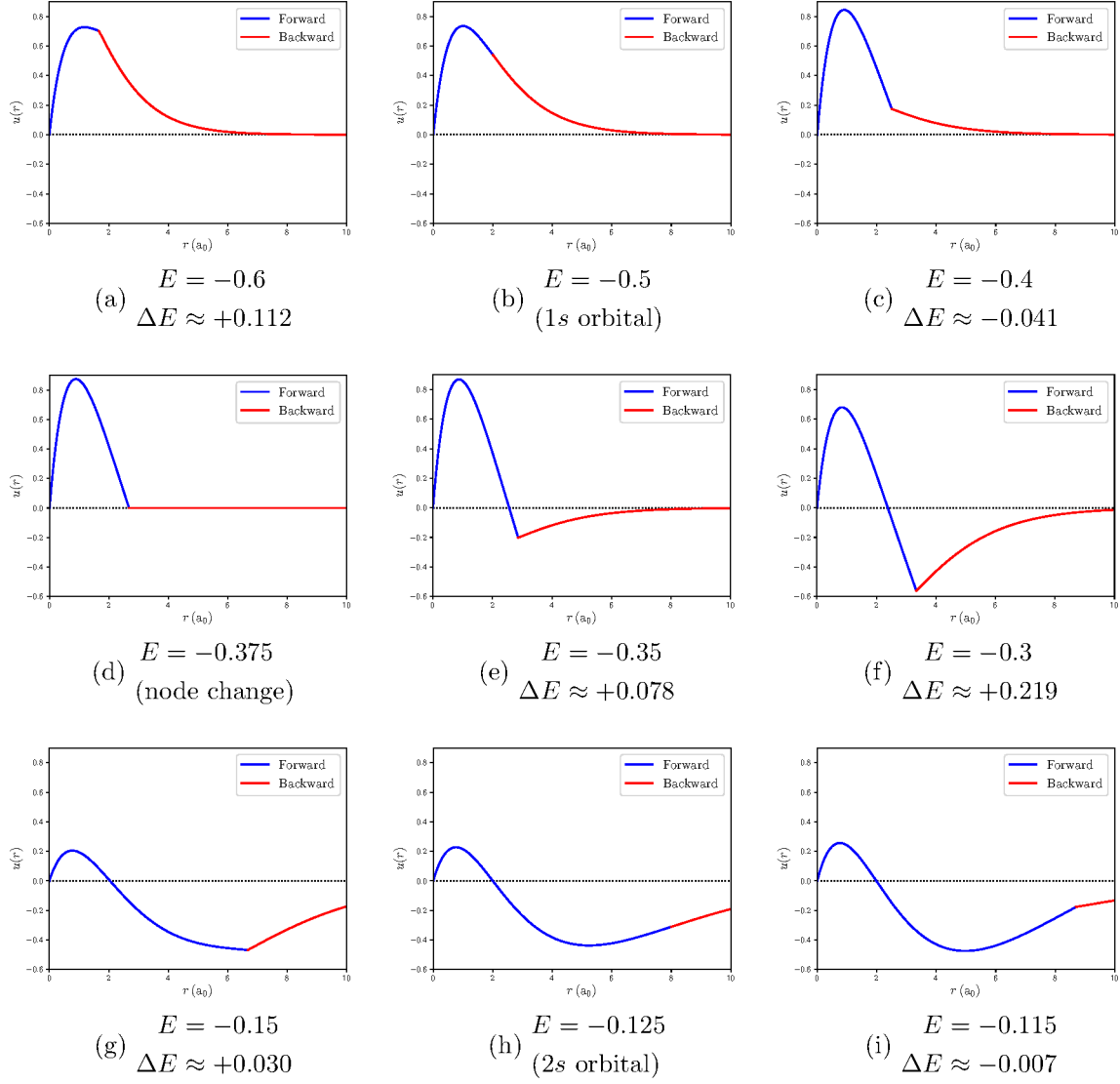


Figure 3.3.: A few testing energies with the corresponding proposed ΔE . The eigen-energies are $E_{1s} = -0.5$ and $E_{2s} = -0.125$. The neighboring energies produce ΔE 's towards the eigen-energies. When the number of nodes changes (around $E = -0.375$), $\Delta E = 0$ occurs, which does not represent an eigen-state. The neighboring ΔE 's do not bring the neighboring energies towards the “fake” root.

3.7. The eigen-state solver

Algorithm 3.2 implemented the basic `shoot` method in the preliminary `Orbit` class. In the test run (Algorithm 3.3), we selected the orbital energy “by hand”. In this section, we provide the complete `Orbit` class (Algorithm 3.4) that implements the perturbation theory and solves for the eigen-state automatically.

Searching for the eigen-energy is a root-finding process. Ideally, if the guessed energy is close enough to the eigen-energy, by simple iterations, the solution will converge automatically to the eigen-state. This simple algorithm is implemented in the method `shootrepeat`. It takes an initial guess of the orbital energy `Ei`, and iterates a few times trying to converge ΔE to a given tolerance. Certainly, `shootrepeat` only works if `Ei` is already close to the root. It doesn’t work for a random guess. Later, when we solve for the self-consistent field problem, `shootrepeat` plays an important role which dramatically improves the root-finding efficiency. Because during the self-consistent field loop, the eigen-energies of the orbitals in the current iteration are usually very close to the eigen-energies from the pervious step, which are the ideal input values for `Ei`.

In general, to solve for a single eigen-state, we should provide an energy window (or bracket) that contains a single eigen-energy. Then we let the window shrink monotonically to the solution. Looking for a proper energy bracket is the first important step. To determine the energy bracket, we need to check the number of nodes in the wave function and the corresponding ΔE . A node is a point along the wave function where the function value goes through zero. On a discrete grid, a node happens when $u_i * u_{i+1} < 0$. In Python, counting the number of nodes can be easily implemented in one line: `node = numpy.sum(u[:-1]*u[1:]<0)`. The condition `u[:-1]*u[1:]<0` returns an array of booleans, where a true value represents a node. And the `numpy.sum` counts the number of `True`’s. For a given principal quantum number n and angular quantum number l , the number of nodes satisfies,

$$node = n - l - 1 \quad (3.31)$$

For instance, $1s$, $2p$, $3d$ orbitals have 0 node; $2s$, $3p$, $4d$ orbitals have 1 node; $3s$, $4p$, $5d$ orbitals have 2 nodes; etc.. For an energy that produces the correct number of nodes in the wave function, if $\Delta E > 0$, perturbation theory suggests the testing energy to move higher; or if $\Delta E < 0$, it suggests the testing energy to move lower. Therefore, a valid energy bracket (`Em`, `Ep`) should satisfy the following condition:

$$\begin{aligned} \text{shoot}(\text{Em}) &\rightarrow u \rightarrow \text{node} == n - l - 1 \text{ and } dE > 0 \\ \text{shoot}(\text{Ep}) &\rightarrow u \rightarrow \text{node} == n - l - 1 \text{ and } dE < 0 \end{aligned}$$

Analytically, we know that for hydrogen-like systems, “all” the eigen-energies are bounded in between $[-\frac{Z^2}{2}, 0)$, where $-\frac{Z^2}{2}$ is the ground state energy and 0 is the boundary between bound and scattering states. Even when later we introduce the Kohn-Sham potential, $[-\frac{Z^2}{2}, 0)$ will still be the global energy bound. This is the starting interval in which we search for the small energy bracket (we might go a bit beyond in case of some numerical artifacts). Solving for the eigen-state is implemented in the `solve` method. A bisection

3. Self-consistent field solver for atomic systems

scheme is used to search for the energy bracket. After a proper energy bracket is found, the program iteratively shoots, updates the orbital energy, and shrinks the bracket, with a catch that if the updated energy goes beyond the bracket then the program rejects the proposed ΔE and simply shrinks the bracket by half. Thanks to the fast convergence rate of the first-order perturbation scheme, typically, once a proper energy bracket is found, it takes less than ten iteration steps to converge $|\Delta E| < 10^{-8}$.

At this stage, creating an atomic wave function should be super easy. Once an `Orbit` object is created, by default it calls the `solve` method implicitly. The numerical solution will be generated automatically for the given potential on the grid. A testing `run.py` is shown in Algorithm 3.5. We generate hydrogen wave functions $1s$, $2s$, $2p$, $3s$, $3p$, and $3d$, and the results are plotted in Fig. 3.4.

Algorithm 3.4: The complete `Orbit` class (`orbit.py`).

```

1  import math
2  import numpy as np
3  import scipy.integrate as sp
4
5  class Orbit:
6      def __init__(self, G, n, l, occ=0, Ei=None, isSolved=True):
7          self.G = G          # Grid
8          self.n = n          # Principal quantum number
9          self.l = l          # Angular momentum quantum number
10         self.occ = occ       # Number of occupied electrons
11         self.R = np.zeros(G.N) # Radial wave function
12         self.u = np.zeros(G.N) # u = r*R
13         self.E = None        # Orbital energy
14         self.M = None        # Matching point index
15         self.node = None     # Number of nodes
16         self.dE = None       # Proposed dE from perturbation
17         if isSolved: self.solve(Ei) # Return the solved orbital
18
19     # Shoot a wave function with given E
20     def shoot(self, E):
21         # Collect attributes (overwrite u to re-use the memory space)
22         (N, dx, r, V, l, u) = (
23             self.G.N, self.G.dx, self.G.r, self.G.V, self.l, self.u
24         )
25
26         # Matching point
27         M = np.where(E<V)[0][0] if E<V[-2] else N//2
28
29         # During the calculation, u is the rescaled u/sqrt(r)
30         # Forward initial condition [0][1]-->
31         u[:2] = r[:2]**(1+0.5)
32         # Backward initial condition <--[end-1][end]
33         end = N-1
34         if E<V[-2]:
35             # Exponential decay, avoid numerical underflow
36             tail = np.exp(-math.sqrt(abs(2*E))*r)/np.sqrt(r)
37             end = np.where(tail>1e-16)[0][-1] # After [end], tail too small
38             u[end-1:] = tail[end-1:]
39         else:
40             # Energy too high, use hard boundary condition
41             u[-2:] = [1e-8, 0.0]
42
43         # Numerov method
44         kk = 2*(r**2*(E-V) - 0.5*(1+0.5)**2)
45         A = 2 - dx*dx*5/6*kk
46         B = 1 + dx*dx/12*kk

```

```

47     # Forward integration [0][1]-->[M]
48     for i in range(2, M+1):
49         u[i] = (A[i-1]*u[i-1] - B[i-2]*u[i-2]) / B[i]
50     FM = u[M]
51     # Backward integration [M]<--[end-1][end]
52     for i in range(end-2, M-1, -1):
53         u[i] = (A[i+1]*u[i+1] - B[i+2]*u[i+2]) / B[i]
54     BM = u[M]
55     # Connect forward and backward parts
56     u[M:] *= FM/BM
57
58     # Normalization
59     u /= math.sqrt(sp.simps(r**2*u**2, dx=dx))
60
61     # Count the number of nodes
62     node = np.sum(u[:-1]*u[1:]<0)
63
64     # Propose dE from perturbation (0.834 is an empirical speed up factor)
65     dE = ((6/5/dx - kk[M] /2 *dx)*u[M]
66           -(3/5/dx + kk[M+1]/20*dx)*u[M+1]
67           -(3/5/dx + kk[M-1]/20*dx)*u[M-1])*u[M]*0.834
68
69     # Collect results and rescale u back to u = r*R
70     (self.R, self.u, self.E, self.node, self.M, self.dE) = (
71         u/np.sqrt(r), u*np.sqrt(r), E, node, M, dE
72     )
73
74     # Shoot repeatedly from Ei until dE converges
75     def shootrepeat(self, Ei, itMax=10):
76         self.shoot(Ei)
77         for it in range(itMax):
78             # Update E by perturbation theory
79             E = self.E + self.dE
80             # Shoot and determine the dE
81             self.shoot(E); dE = self.dE
82             if abs(dE)<1e-8:
83                 return True # Converged
84         return False
85
86     # Solve the eigen-state
87     def solve(self, Ei=None):
88         # Collect attributes
89         (Z, n, l) = (self.G.Z, self.n, self.l)
90
91         # Try Ei if available
92         if Ei is not None:
93             with np.errstate(all='ignore'):
94                 res = self.shootrepeat(Ei)
95                 if res and self.node==n-1-1: # Lucky Ei, problem solved
96                     return True
97
98         # Otherwise, find an energy bracket and solve the problem
99         # Initial bracket
100         (Em, Ep) = (-0.55*Z**2, 0.05*Z**2)
101
102         # Error message
103         errmsg = "Orbit: n=%d, l=%d not found in (%.2f, %.2f)\n" % (n, l, Em, Ep)
104
105         # Check lower and upper bound
106         self.shoot(Em); (ndm, dEm) = (self.node, self.dE)
107         self.shoot(Ep); (ndp, dEp) = (self.node, self.dE)
108         if ndm>n-1-1 or ndp<n-1-1 or (ndm==n-1-1 and dEm<0) or (ndp==n-1-1 and dEp>0):
109             print(errmsg); return False
110
111         itMax = 100
112         mPass = pPass = False
113         # Find lower bracket (bisection)
114         (em, ep) = (Em, Ep)

```

3. Self-consistent field solver for atomic systems

```

115         for it in range(itMax):
116             E = 0.5*(em+ep)
117             self.shoot(E); (nd, dE) = (self.node, self.dE)
118             if nd==n-1-1 and dE>0:
119                 (Em, mPass) = (E, True); break
120             else:
121                 (em, ep) = (E, ep) if nd<n-1-1 else (em, E)
122         # Find upper bracket (bisection)
123         (em, ep) = (Em, Ep)
124         for it in range(itMax):
125             E = 0.5*(em+ep)
126             self.shoot(E); (nd, dE) = (self.node, self.dE)
127             if nd==n-1-1 and dE<0:
128                 (Ep, pPass) = (E, True); break
129             else:
130                 (em, ep) = (em, E) if nd>n-1-1 else (E, ep)
131
132         if mPass and pPass: # Bracket found, start root finding
133             # Initial shoot
134             self.shoot(0.5*(Em+Ep))
135             # Shoot repeatedly within (Em, Ep) until dE converges
136             for it in range(itMax):
137                 # Update E by perturbation theory
138                 E = self.E + self.dE
139                 # Prevent E going beyond bracket
140                 if E<=Em or E>=Ep:
141                     E = 0.5*(Em+Ep)
142                 # Shoot and determine the dE
143                 self.shoot(E); dE = self.dE
144                 # Check dE and shrink energy bracket
145                 if abs(dE)<1e-8:
146                     return True # Converged
147                 else:
148                     (Em, Ep) = (E, Ep) if dE>0 else (Em, E)
149
150         print(errmsg); return False

```

Algorithm 3.5: A test run.

```

1  from grid import Grid
2  from orbit import Orbit
3  import matplotlib.pyplot as plt
4
5  # Set up grid
6  G = Grid(1, 1e-6, 50, 0.005)
7
8  # Set up orbit
9  orb1s = Orbit(G, 1, 0)
10 orb2s = Orbit(G, 2, 0)
11 orb2p = Orbit(G, 2, 1)
12 orb3s = Orbit(G, 3, 0)
13 orb3p = Orbit(G, 3, 1)
14 orb3d = Orbit(G, 3, 2)
15
16 # Plot
17 plt.plot(G.r, orb1s.u, color='#0000ff', linestyle='-', label='$E_{1s}=%.6f$'%orb1s.E)
18 plt.plot(G.r, orb2s.u, color='#0000ff', linestyle='--', label='$E_{2s}=%.6f$'%orb2s.E)
19 plt.plot(G.r, orb2p.u, color='#ff9900', linestyle='--', label='$E_{2p}=%.6f$'%orb2p.E)
20 plt.plot(G.r, orb3s.u, color='#0000ff', linestyle=':', label='$E_{3s}=%.6f$'%orb3s.E)
21 plt.plot(G.r, orb3p.u, color='#ff9900', linestyle=':', label='$E_{3p}=%.6f$'%orb3p.E)
22 plt.plot(G.r, orb3d.u, color='#ff0000', linestyle=':', label='$E_{3d}=%.6f$'%orb3d.E)
23 plt.legend(loc='best')
24 plt.show()

```

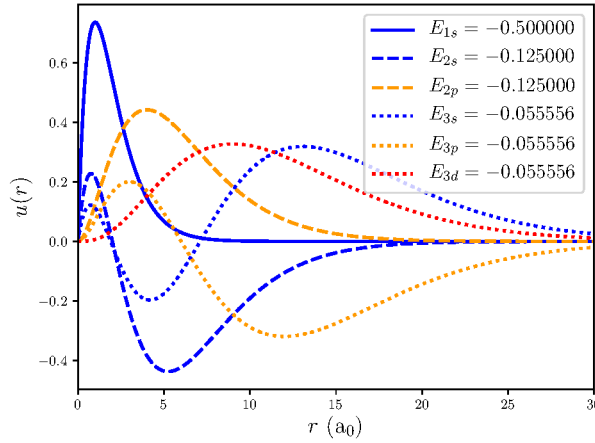


Figure 3.4.: Numerical eigen-functions and eigen-energies of hydrogen 1s, 2s, 2p, 3s, 3p, and 3d orbitals.

3.8. Many-electron atomic system and the Kohn-Sham equation

From the previous few sections, we have developed a program to solve hydrogen-like systems, where only one electron is present. In the framework of density functional theory (DFT), we are going to generalize our program to solve many-electron atomic systems.

The N_e -electron Schrödinger equation (in a.u.) reads,

$$\left\{ \sum_{i=1}^{N_e} \left[-\frac{1}{2} \nabla_i^2 + V_{\text{ext}}(\mathbf{r}_i) \right] + \sum_{i<j}^{N_e} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \right\} \Psi = E \Psi \quad (3.32)$$

The philosophy of DFT is that any property of a many-body system can be viewed as a *functional* of the ground state electron *density* $\rho(\mathbf{r})$. The existence for such functionals is proved in the original works by Hohenberg and Kohn [11]. There was, however, no guidance for constructing or approximating the functionals, until the works extended by Kohn and Sham [12]. The spirit of the Kohn-Sham approach is to replace the original many-body problem by an auxiliary problem which is a set of N_e one-electron Schrödinger-like equations, known as the Kohn-Sham equations [12]:

$$\left[-\frac{1}{2} \nabla^2 + V_{\text{KS}}(\mathbf{r}) \right] \varphi_i = \varepsilon_i \varphi_i \quad \text{for } i = 1, 2, \dots, N_e \quad (3.33)$$

where $V_{\text{KS}}(\mathbf{r})$ is a one-electron effective mean-field potential, called the Kohn-Sham potential. It consists of an external potential $V_{\text{ext}}(\mathbf{r})$, an Hartree potential $V_{\text{Hartree}}(\mathbf{r})$, and an exchange-correlation potential $V_{\text{xc}}(\mathbf{r})$:

$$V_{\text{KS}}(\mathbf{r}) = V_{\text{ext}}(\mathbf{r}) + V_{\text{Hartree}}(\mathbf{r}) + V_{\text{xc}}(\mathbf{r}) \quad (3.34)$$

3. Self-consistent field solver for atomic systems

φ_i and ε_i are the Kohn-Sham orbitals and energies. The “soul” of DFT, the electron density, is given by:

$$\rho(\mathbf{r}) = \sum_{i=1}^{N_e} |\varphi_i(\mathbf{r})|^2 \quad (3.35)$$

Consider a carbon atom (${}_6\text{C}$) with the electronic configuration $1s^2, 2s^2, 2p^2$. The $1s$ and $2s$ shells are completely filled, while the $2p$ shell (which can be filled with up to 6 electrons) is only partially filled with 2 electrons. It is not given – from the electronic configuration – which 2 of the 6 orbitals are occupied. Here we introduce the “fractional occupations” that *each* electron occupies *every* orbital in the shell fractionally and evenly. This has the advantage and simplicity that the *spherical symmetry* of the electron density is preserved.

Proof:

We want to prove that, for a given shell, if the orbitals are evenly occupied, the corresponding electron density is spherically symmetric: $\rho(\mathbf{r}) \rightarrow \rho(r)$.

Use the “addition theorem” of spherical harmonics: ($\hat{\mathbf{r}}_1$ and $\hat{\mathbf{r}}_2$ are unit vectors, P_l are Legendre polynomials)

$$P_l(\hat{\mathbf{r}}_1 \cdot \hat{\mathbf{r}}_2) = \frac{4\pi}{2l+1} \sum_{m=-l}^l \overline{Y_{lm}(\theta_1, \phi_1)} Y_{lm}(\theta_2, \phi_2) \quad (3.36)$$

If $\hat{\mathbf{r}}_1 = \hat{\mathbf{r}}_2 = \hat{\mathbf{r}}$, we have,

$$P_l(1) = 1 = \frac{4\pi}{2l+1} \sum_{m=-l}^l |Y_{lm}(\theta, \phi)|^2 \quad (3.37)$$

The electron density contributed from each electron reads,

$$\rho(\mathbf{r}) = |R_{nl}(r)|^2 \underbrace{\left(\frac{1}{2l+1} \sum_{m=-l}^l |Y_{lm}(\theta, \phi)|^2 \right)}_{\text{angular dependence averaged out}} = \frac{1}{4\pi} |R_{nl}(r)|^2 \quad \text{Q.E.D.} \quad (3.38)$$

By introducing the “fractional occupations”, the electron density becomes spherically symmetric:

$$\rho(r) = \frac{1}{4\pi} \sum_{i=1}^{N_e} |R_i(r)|^2 \quad (3.39)$$

For the atomic systems, the external potential $V_{\text{ext}}(r) = -Z/r$ is, by itself, spherically symmetric. The Hartree and the exchange-correlation potentials depend on the distribution of the electron density, which is now spherically symmetric. As a result, we have a spherically symmetric Kohn-Sham potential $V_{\text{KS}}(r) = V_{\text{ext}}(r) + V_{\text{Hartree}}(r) + V_{\text{xc}}(r)$. This spherical-distribution approximation brings the Kohn-Sham equation (3.33) to a form

that is similar to the hydrogen-like system (3.1). Thus, if the Hartree and the exchange-correlation potentials are given, we can easily extend our hydrogen-like atomic solver to solve general many-electron atomic systems.

3.9. Hartree and exchange-correlation potentials

The Hartree potential is perhaps the most significant component in describing the electron-electron interactions, as it represents the classical Coulomb repulsion by picturing the electron density as a continuous charge distribution.

For a given *spherically symmetric* electron density $\rho(r)$, the total charge (in units of $-e$) enclosed in a sphere with radius r can be simply integrated as

$$Q(r) = \int_0^{2\pi} d\phi \int_0^\pi d\theta \sin \theta \int_0^r dr' r'^2 \rho(r') = 4\pi \int_0^r dr' r'^2 \rho(r') \quad (3.40)$$

Setting the zero potential reference point at infinity, the electric potential generated by the charge is [17]: (in atomic units)

$$V_{\text{Hartree}}(r) = \int_r^\infty dr' \frac{Q(r')}{r'^2} \quad (3.41)$$

Numerically, the enclosed charge Q reaches the number of electrons N_e at r_{max} , since this is how the wave functions are normalized. On a finite grid, the Hartree potential becomes:

$$V_{\text{Hartree}}(r) = \int_r^{r_{\text{max}}} dr' \frac{Q(r')}{r'^2} + \int_{r_{\text{max}}}^\infty dr' \frac{N_e}{r'^2} = \int_r^{r_{\text{max}}} dr' \frac{Q(r')}{r'^2} + \frac{N_e}{r_{\text{max}}} \quad (3.42)$$

Hartree potential and energy:

A more direct formula connecting the charge density and the electric potential is [17]:

$$V_{\text{Hartree}}(\mathbf{r}) = \int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (3.43)$$

The corresponding energy reads [17]:

$$E_{\text{Hartree}} = \frac{1}{2} \int d\mathbf{r} \int d\mathbf{r}' \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (3.44)$$

Those formalisms are more useful when $\rho(\mathbf{r})$ has an arbitrary distribution. However, computing (3.43) and (3.44) can be formidable even for simple charge distributions. In our problem setting, the ρ is spherically symmetric. That is the ideal symmetry for applying Gauss's law [17] and obtain the electric potential (3.41) with little effort.

3. Self-consistent field solver for atomic systems

The Hartree potential describes the classical electron-electron interaction in a mean-field picture. The remaining quantum mechanical many-body effects on the density are described via the exchange-correlation potential, given as a functional derivative:

$$V_{\text{xc}}(\mathbf{r}) = \frac{\delta E_{\text{xc}}[\rho]}{\delta \rho(\mathbf{r})} \quad (3.45)$$

where E_{xc} is the exchange-correlation energy, which is a *universal* but *unknown* functional of the density ρ . Frequently used approximations for the density functional $E_{\text{xc}}[\rho]$ are the local density approximation (LDA), the local spin density approximation (LSDA), and the generalized gradient approximations (GGA). Reference [18] gives an excellent summary and discussion on the various approximations.

In the relatively simple atomic systems, our choice is to use the LDA for the exchange-correlation potential approximation. In LDA, the functionals are assumed to be dependent solely upon the value of ρ at each point in space (but not, for example, derivatives of ρ). In general, in LDA, the exchange-correlation energy is written as

$$E_{\text{xc}}^{\text{LDA}}[\rho] = \int d\mathbf{r} \rho(\mathbf{r}) \epsilon_{\text{xc}}(\rho(\mathbf{r})) \quad (3.46)$$

where ϵ_{xc} is the exchange-correlation energy density derived from the homogeneous electron gas (HEG) model. For an HEG with electron density ρ (a constant), the exchange-correlation energy density ϵ_{xc} is determined by ρ . In this regard, ϵ_{xc} is a function of ρ . In realistic systems, $\rho(\mathbf{r})$ is no longer a constant. However, we assume that $\rho(\mathbf{r})$ is slow varying, the corresponding exchange-correlation energy density can be taken from the HEG model with the corresponding electron density: $\epsilon_{\text{xc}}(\rho(\mathbf{r}))$.

The exchange-correlation energy density is decomposed into an exchange term and a correlation term linearly:

$$\epsilon_{\text{xc}} = \epsilon_{\text{x}} + \epsilon_{\text{c}} \quad (3.47)$$

The exchange term ϵ_{x} is given by the Kohn-Sham-Gasp r approximation [19]:

$$\epsilon_{\text{x}} = - \left(\frac{3}{4\pi} \right) \frac{1}{\alpha r_s} \quad (3.48)$$

where,

$$r_s = \left(\frac{3}{4\pi\rho(\mathbf{r})} \right)^{\frac{1}{3}} \quad \text{and} \quad \alpha = \left(\frac{4}{9\pi} \right)^{\frac{1}{3}}$$

The correlation term ϵ_{c} , on the other hand, has a number of different approximations with different parameterizations. A detailed discussion can be found in Reference [19]. Here we provide the Ceperley-Alder approximation with Vosko-Wilk-Nusair parameterization (CA-VWN) which is the LDA recommended by Reference [19]:

$$\epsilon_{\text{c}} = A \left[\ln \frac{r_s}{X(r_s)} + \frac{2b}{Q} \tan^{-1} \frac{Q}{2\sqrt{r_s+b}} - \frac{bx_0}{X(x_0^2)} \left(\ln \frac{(\sqrt{r_s}-x_0)^2}{X(r_s)} + \frac{2(b+2x_0)}{Q} \tan^{-1} \frac{Q}{2\sqrt{r_s+b}} \right) \right] \quad (3.49)$$

with,

$$X(r_s) = r_s + b\sqrt{r_s} + c \quad \text{and} \quad Q = \sqrt{4c - b^2}$$

The other parameters are given as:

A	0.0310907
b	3.72744
c	12.9352
x_0	-0.10498
b_1	9.81379
b_2	2.82224
b_3	0.736412

Those are the “paramagnetic” parameters³ tabulated in Reference [19], where the up-spin’s and down-spin’s are treated equally (preserving the spin symmetry), in contrast to the local spin density approximation (LSDA).

Having the expression of ϵ_{xc} as the sum of (3.48) and (3.49), one can obtain the exchange-correlation *potential* via the functional derivative (3.45):

$$V_{\text{xc}}(\mathbf{r}) = \frac{\delta}{\delta \rho(\mathbf{r})} \left[\rho(\mathbf{r}) \epsilon_{\text{xc}}(\rho(\mathbf{r})) \right] \quad (3.50)$$

which yields,

$$V_{\text{xc}}(\mathbf{r}) = \frac{4}{3} \epsilon_{\text{x}} + \epsilon_{\text{c}} - \frac{A}{3} \frac{1 + b_1 r_s^{1/2}}{1 + b_1 r_s^{1/2} + b_2 r_s + b_3 r_s^{3/2}} \quad (3.51)$$

Eqn. (3.51) is the final expression for the rather complicated exchange-correlation potential in LDA with the Ceperley-Alder approximation and the Vosko-Wilk-Nusair parameterization (CA-VWN). As a remark, the definition of r_s (known as the Wigner-Seitz radius) contains a division by the electron density ρ , which involves a division-by-zero risk when the electron density is zero. Numerically, when $\rho = 0$, we do not need to perform a calculation since we know that $V_{\text{xc}} = 0$.

3.10. Total energy of the many-body system

The formalism of the Kohn-Sham equation (3.33) is built on minimizing the total energy – the key physical quantity – of the many-body system. The total energy is given as a sum of 4 energy terms: the kinetic energy E_{kin} , the external potential energy E_{ext} , the Hartree energy E_{Hartree} , and the exchange-correlation energy E_{xc} :

$$E_{\text{tot}} = E_{\text{kin}} + E_{\text{ext}} + E_{\text{Hartree}} + E_{\text{xc}} \quad (3.52)$$

³The parameter A is given in units of Hartree. It differs from the number in Reference [19] by a factor of 2, which is given in units of Rydberg.

3. Self-consistent field solver for atomic systems

where,

$$E_{\text{kin}} = \sum_{i=1}^{N_e} \langle i | -\frac{1}{2} \nabla^2 | i \rangle \quad (3.53)$$

$$E_{\text{ext}} = \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{ext}}(\mathbf{r}) \quad (3.54)$$

$$E_{\text{Hartree}} = \frac{1}{2} \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{Hartree}}(\mathbf{r}) \quad (3.55)$$

$$E_{\text{xc}} = \int d\mathbf{r} \rho(\mathbf{r}) \epsilon_{\text{xc}}(\mathbf{r}) \quad (3.56)$$

Things might be more subtle than they appear to be. There are a few issues that I would like to discuss:

1. The sum of the individual Kohn-Sham orbital energies does *not* yield the total energy of the system:

$$E_{\text{tot}} \neq \sum_{i=1}^{N_e} \varepsilon_i$$

The Kohn-Sham orbital energies are solutions to the auxiliary non-interacting problems. They do not add up to the total energy of the many-electron system.

2. The kinetic energy E_{kin} , as given in (3.53), is the sum of the non-interacting particles' kinetic energies. It is *not* the kinetic energy of the many-body system. In the Kohn-Sham approach, this discrepancy is absorbed into the exchange-correlation term.
3. E_{ext} is the external potential energy of the many-body system. E_{ext} can be viewed as a linear functional of ρ , as shown in (3.54), because the external potential V_{ext} is independent of ρ (as the name implies).
4. E_{Hartree} is the Hartree energy of the many-body system. E_{Hartree} can be viewed as a quadratic functional of ρ , as explicitly written in (3.44). The $\frac{1}{2}$ prefactor accounts for the electron-pair double counting, it can also be seen from the functional derivative perspective $V_{\text{Hartree}} = \delta E_{\text{Hartree}} / \delta \rho$ connecting (3.43) and (3.44).
5. E_{xc} is the exchange-correlation energy of the many-body system. In the original definition (3.46), the integral over the electron density and the energy density produces the energy. We do not write (3.56) in terms of V_{xc} , because the functional derivative that relates V_{xc} and ϵ_{xc} does not lead to a simple polynomial relation.

One can avoid taking derivatives (which involves unnecessary numerical errors) in the kinetic term (3.53), by considering the sum of the Kohn-Sham orbital energies:

$$\begin{aligned} \sum_{i=1}^{N_e} \varepsilon_i &= \sum_{i=1}^{N_e} \langle i | -\frac{1}{2} \nabla^2 + V_{\text{ext}} + V_{\text{Hartree}} + V_{\text{xc}} | i \rangle \\ &= E_{\text{kin}} + E_{\text{ext}} + 2E_{\text{Hartree}} + \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{xc}}(\mathbf{r}) \end{aligned} \quad (3.57)$$

from which we can compute E_{kin} in a different angle:

$$E_{\text{kin}} = \sum_{i=1}^{N_e} \varepsilon_i - E_{\text{ext}} - 2E_{\text{Hartree}} - \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{xc}}(\mathbf{r}) \quad (3.58)$$

As a result, comparing with (3.52), the total energy can be better computed numerically as:

$$E_{\text{tot}} = \sum_{i=1}^{N_e} \varepsilon_i - E_{\text{Hartree}} - \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{xc}}(\mathbf{r}) + E_{\text{xc}} \quad (3.59)$$

3.11. The self-consistent field solver

We are now in a position to solve the complete problem self-consistently. The self-consistent loop is shown in Fig. 3.5. The loop starts from an initial potential, then continues solving the Kohn-Sham equation and updating the potential until the solution converges.

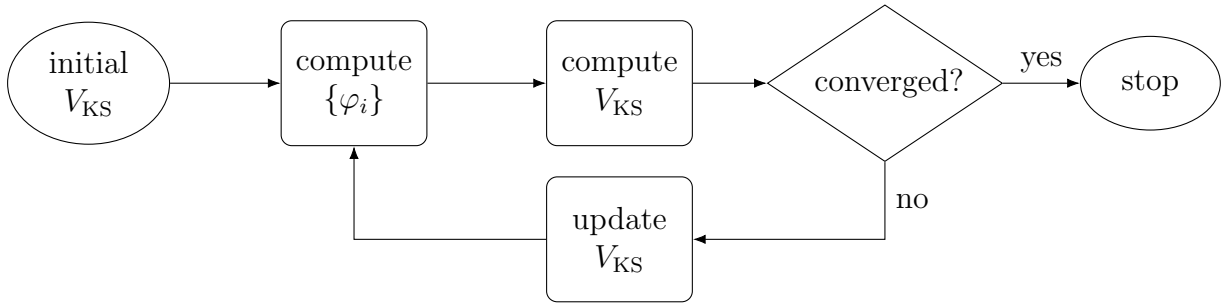


Figure 3.5.: Flow chart of self-consistent field iteration. The loop starts from an initial potential, then continues solving the Kohn-Sham equation and updating the potential until the solution converges.

1. Initially, we set $V_{\text{Hartree}} = 0$ and $V_{\text{xc}} = 0$.

$$V_{\text{KS}} \leftarrow V_{\text{ext}}$$

This assumes that the electrons are as if non-interacting, which produces hydrogen-like orbitals as our very first result.

2. From the produced electron density, we compute V_{Hartree} and V_{xc} according to (3.42) and (3.51), and update V_{KS} by a linear-mixing:

$$V_{\text{KS}} \leftarrow (1 - \alpha)V_{\text{KS}} + \alpha(V_{\text{ext}} + V_{\text{Hartree}} + V_{\text{xc}})$$

where α is a number between 0 and 1. If $\alpha = 1$, it is a full update, which normally makes the problem non-converging. Practically, we choose $\alpha = 0.5$. However, that still does not guarantee a convergence. During the iterations, if the solution starts to diverge, we fine-tune α towards 0.3 to ensure the convergence.

3. Self-consistent field solver for atomic systems

3. From the updated potential V_{KS} , we solve again the Kohn-Sham equation and obtain new orbitals and the corresponding electron density. The loop continues until the solutions are converged.
4. Finally, we output the physical quantities, including the Kohn-Sham orbitals, the electron density, the potentials, and the total energy of the system.

The complete self-consistent field solver is given in Algorithm 3.6.

Algorithm 3.6: The self-consistent field solver SCF for atomic systems (`scf.py`).

```
1 import math
2 import numpy as np
3 import scipy.integrate as sp
4 from grid import Grid
5 from orbit import Orbit
6
7 # Cumulatively integrate y using the Simpson's rule
8 def cumsimps(y, dx):
9     I = np.zeros(len(y))
10    I[1] = (y[0]+y[1])/2
11    for i in range(2,len(y)):
12        I[i] = I[i-2] + (y[i-2]+4*y[i-1]+y[i])/3
13    return dx*I
14
15 class SCF:
16     def __init__(self, Z, rmin, rmax, dx, cfgList):
17         G = Grid(Z, rmin, rmax, dx); (r, N) = (G.r, G.N)
18         self.G = G # Grid
19         self.cfgList = cfgList # List of configurations
20         self.orbList = [] # List of orbitals
21         self.EiList = [-Z**2/(2*cfg[0]**2) for cfg in cfgList] # List of Ei
22         self.Ne = np.sum(cfgList, axis=0)[2] # Number of electrons
23         self.Vext = -Z/r # Nuclear potential
24         self.VH = np.zeros(N) # Hartree potential
25         self.Vxc = np.zeros(N) # Exchange-correlation potential
26         self.rho = np.zeros(N) # Electron density
27         self.Etot = 0 # Total energy
28         self.Ekin = 0 # Kinetic energy
29         self.Eext = 0 # Nuclear potential energy
30         self.EH = 0 # Hartree energy
31         self.Exc = 0 # Exchange-correlation energy
32         self.Niter = 0 # Current iteration number
33         self.mix = 0.5 # Linear mixing coefficient
34         self.cnvg = None # Current convergence
35
36     # One SCF step
37     def step(self):
38         # Collect attributes
39         (G, N, dx, r, V, cfgList, EiList, Ne, Vext, mix) = (
40             self.G, self.G.N, self.G.dx, self.G.r, self.G.V,
41             self.cfgList, self.EiList, self.Ne, self.Vext, self.mix
42         )
43
44         # Generate occupied orbitals
45         orbList = [Orbit(G, *cfg, Ei=Ei) for (cfg, Ei) in zip(cfgList, EiList)]
46
47         # Update Ei
48         EiList = [orb.E for orb in orbList]
49
50         # Compute rho from orbList
51         rho = sum(orb.occ*orb.R**2 for orb in orbList)/(4*math.pi)
52
```

```

53     # Charge
54     Q = 4*math.pi*cumsimps(rho*r**3, dx)
55
56     # Hartree potential
57     VH = cumsimps((Q/r)[::-1], dx)[::-1] + Ne/r[-1]
58
59     # Exchange-correlation potential (Ceperley-Alder Vosko-Wilk-Nusair)
60     alpha = (4/(9*math.pi))**(1/3)
61     nz = rho>1e-16 # Avoid division-by-zero
62     rs = (3/(4*math.pi*rho[nz]))**(1/3)
63     (A, b, c, x0, b1, b2, b3) = (
64         0.0310907, 3.72744, 12.9352, -0.10498, 9.81379, 2.82224, 0.736412
65     )
66     Q0 = math.sqrt(4*c-b**2); rsrt = np.sqrt(rs); Xr = rs+b*rsrt+c
67     ex = -0.75/(math.pi*alpha*rs)
68     ec = A*(
69         np.log(rs/Xr) + 2*b/Q0*np.arctan(Q0/(2*rsrt+b)) - b*x0/(x0**2+b*x0+c)*(
70             np.log((rsrt-x0)**2/Xr) + 2*(b+2*x0)/Q0*np.arctan(Q0/(2*rsrt+b))
71         )
72     )
73     exc = np.zeros(N) # Exchange-correlation energy density
74     exc[nz] = ex + ec
75     Vxc = np.zeros(N) # Exchange-correlation potential
76     Vxc[nz] = 4/3*ex + ec - A/3*(1+b1*rsrt)/(1+b1*rsrt+b2*rs+b3*rsrt**3)
77
78     # Total energy
79     Eext = 4*math.pi*sp.simps(r**3*rho*Vext, dx=dx)
80     EH = 2*math.pi*sp.simps(r**3*rho*VH, dx=dx)
81     Evxc = 4*math.pi*sp.simps(r**3*rho*Vxc, dx=dx)
82     Exc = 4*math.pi*sp.simps(r**3*rho*exc, dx=dx)
83     Eig = sum(orb.occ*orb.E for orb in orbList)
84     Ekin = Eig - Eext - 2*EH - Evxc
85     Etot = Eig - EH - Evxc + Exc
86
87     # Convergency
88     cnvg = abs(Etot-self.Etot)
89     if self.cnvg is not None and cnvg>=self.cnvg and mix>0.3:
90         mix -= 0.1 # Reduce mixing if cnvg increases
91
92     # Update V by mixing
93     V = (1-mix)*V + mix*(Vext+VH+Vxc)
94
95     # Collect results
96     (self.G.V, self.orbList, self.EiList, self.VH, self.Vxc, self.rho,
97      self.Etot, self.Ekin, self.Eext, self.EH, self.Exc, self.Niter,
98      self.mix, self.cnvg) = (V, orbList, EiList, VH, Vxc, rho, Etot,
99      Ekin, Eext, EH, Exc, self.Niter+1, mix, cnvg)
100
101     # Many SCF steps until convergence
102     def full(self, eps=1e-8, itMax=100):
103         for it in range(itMax):
104             self.step()
105             if self.cnvg<=eps:
106                 return True
107         return False

```

Simpson's cumulative integral:

We made a helper function `cumsimps` in a similar manner to the `cumtrapz` from the `scipy` package, which helps to compute the integrals (3.40) and (3.42) cumulatively. It is worth to mention that `scipy` provides `trapz`, `simps`, `cumtrapz`, but no `cumsimps`. We could have simply used the trapezoidal rule to compute the Hartree potential, which would be one order less accurate than the Simpson's. However, the Hartree

3. Self-consistent field solver for atomic systems

potential is such a significant term, it should be calculated accurately (there will be a significant difference on the resulting energies, if one replaces `cumsimps` by `cumtrapz`).

In Algorithm 3.7, we perform an SCF calculation for a lead atom ($_{82}\text{Pb}$). Pb has atomic number $Z = 82$, and its electronic configuration is given by (in the Aufbau order):

$$\underbrace{1s^2, 2s^2, 2p^6, 3s^2, 3p^6, 4s^2, 3d^{10}, 4p^6, 5s^2, 4d^{10}, 5p^6, 6s^2, 4f^{14}, 5d^{10}, 6p^2}_{[\text{Xe}]}$$

We run the SCF iterations until the total energy of the system converges to less than a tolerance (10^{-8} Hartree). We plot the initial and final electron density (scaled by r^2) in Fig. 3.6. One can see that, initially, the electron density is concentrated around the nucleus since the electron-electron interactions are ignored. After the SCF calculation, the electron density expands due to the electron-electron interactions and converges to the yellow curve shown in the figure.

Algorithm 3.7: A test run.

```
1 from scf import SCF
2 import matplotlib.pyplot as plt
3
4 # Lead atom (Pb)
5 Z = 82
6 cfgList = [
7     [1, 0, 2],
8     [2, 0, 2],
9     [2, 1, 6],
10    [3, 0, 2],
11    [3, 1, 6],
12    [4, 0, 2],
13    [3, 2, 10],
14    [4, 1, 6],
15    [5, 0, 2],
16    [4, 2, 10],
17    [5, 1, 6],
18    [6, 0, 2],
19    [4, 3, 14],
20    [5, 2, 10],
21    [6, 1, 2]
22 ]
23
24 # Set up SCF solver
25 scf = SCF(Z, 1e-6/Z, 50, 0.005, cfgList); r = scf.G.r
26
27 # SCF calculation
28 scf.step(); rho0 = scf.rho # Initial electron density
29 scf.full(); rho1 = scf.rho # Final electron density
30
31 # Plot
32 plt.plot(r, rho0*r**2, label='Initial_density')
33 plt.plot(r, rho1*r**2, label='SCF_density')
34 plt.legend(loc='best')
35 plt.show()
```

The National Institute of Standards and Technology (NIST) provides excellent references for atomic electronic structure calculations [13]. We compare our results for the lead atom ($_{82}\text{Pb}$) with the LDA results from NIST [13]. The total energy, together with the

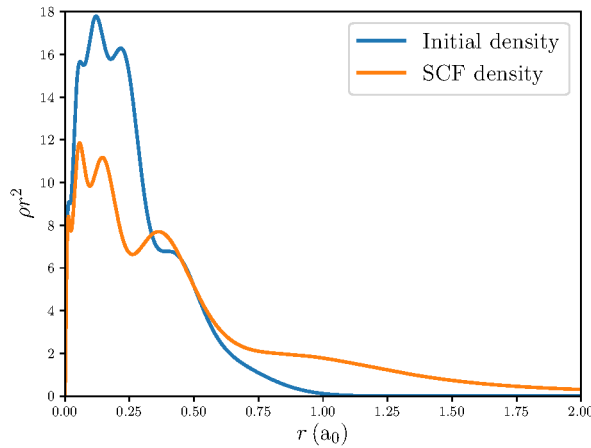


Figure 3.6.: Self-consistent field calculation of a lead atom ($_{82}\text{Pb}$). Blue curve: Initial electron density before SCF iteration; Yellow curve: Final electron density after SCF converges.

kinetic, external potential, Hartree, and exchange-correlation energies, plus the Kohn-Sham orbital energies are tabulated in Table 3.3. We see that the numerical results perfectly agree with each other. With this nicely tabulated results, we conclude this chapter. We have developed a complete, hopefully elegant, DFT code for atomic systems.

A final remark:

Normally, people tend to avoid explicitly listing the source code in the article, since it is often “dangerous”. One has to be very careful not to make a mistake, and to make the code optimized but at the same time easy to read. I am not claiming that I have provided “the best solution”. But I am convinced enough to print out the Python source code, which I believe is carefully organized and conveys precisely the information on the DFT calculation, including the logarithmic grid set up, Numerov’s integration, eigen-value problem solving, Kohn-Sham potential calculation, and self-consistent field iterations. From this explicitly listed Python source code, one can easily run and check the calculations for any different atoms and ions with the chosen electronic configurations.

3. Self-consistent field solver for atomic systems

Table 3.3.: Comparison of SCF calculations between my results and NIST results for a lead atom ($_{82}\text{Pb}$). Energies are given in units of Hartree.

Energy	My results	NIST results
E_{tot}	-19518.993145	-19518.993145
E_{kin}	19512.662680	19512.662684
E_{ext}	-46459.627739	-46459.627742
E_{Hartree}	7781.994460	7781.994459
E_{xc}	-354.022546	-354.022546
$1s^2$	-2901.078061	-2901.078061
$2s^2$	-488.843335	-488.843335
$2p^6$	-470.877785	-470.877785
$3s^2$	-116.526852	-116.526852
$3p^6$	-107.950391	-107.950391
$3d^{10}$	-91.889924	-91.889924
$4s^2$	-25.753330	-25.753330
$4p^6$	-21.990564	-21.990564
$4d^{10}$	-15.030027	-15.030026
$4f^{14}$	-5.592532	-5.592532
$5s^2$	-4.206798	-4.206797
$5p^6$	-2.941657	-2.941657
$5d^{10}$	-0.902393	-0.902393
$6s^2$	-0.357187	-0.357187
$6p^2$	-0.141831	-0.141831

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

4.1. Atomic open-shell problems

The “spherical charge-distribution approximation” (3.39) that we made in the DFT calculation last chapter was *exact* for closed shells. Eqn. (3.39) is an approximation only for shells that are not completely filled – we call them open-shells.

Open-shell problems are many-body problems.¹ In this chapter, we are going to solve the atomic open-shell systems and discuss the many-body effects of the two important interactions:

1. The Coulomb Hamiltonian (in atomic units)

$$\text{1st quantization: } H_U = \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \quad (4.1)$$

$$\text{2nd quantization: } H_U = \sum_{\substack{\alpha < \beta \\ \gamma < \delta}} \ddot{U}_{\alpha\beta\gamma\delta} c_\alpha^\dagger c_\beta^\dagger c_\gamma c_\delta; \quad \ddot{U}_{\alpha\beta\gamma\delta} = \langle \alpha\beta | \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} | \gamma\delta \rangle \quad (4.2)$$

(the form of the two-electron integral was discussed in (1.19))

2. The spin-orbit Hamiltonian (in atomic units)

$$\text{1st quantization: } H_{\text{SO}} = \sum_{i=1}^{N_e} \xi(r_i) \boldsymbol{\ell}_i \cdot \mathbf{s}_i; \quad \xi(r) = \frac{1}{2c^2 r} \frac{dV}{dr} \quad (4.3)$$

$$\text{2nd quantization: } H_{\text{SO}} = \sum_{\alpha\beta} X_{\alpha\beta} c_\alpha^\dagger c_\beta; \quad X_{\alpha\beta} = \langle \alpha | \xi(r) \boldsymbol{\ell} \cdot \mathbf{s} | \beta \rangle \quad (4.4)$$

where, the speed of light is $c \approx 137.036 \, a_0/t_0$ in atomic units; and $V(r)$ is the mean-field potential in the one-electron picture.

¹Of course, closed shell problems are also “many-body” since there are many electrons. But closed shells (1-dim Hilbert space) can be reduced effectively to one-electron problems. They are not the “many-body problem” in our sense.

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

Atomic open-shells are precious systems in quantum *many-body* problems that can have *analytic solutions*. In this chapter, we discuss how to diagonalize the many-electron Hamiltonians H_U and H_{SO} in the open-shell systems *analytically*, without setting up the matrix representations. If you think about it, this is very cool: *we are going to diagonalize matrices without using the matrices*. This is only possible because the Hamiltonians have special symmetries: they commute with certain angular momentum operators, which allows us to use the **ladder operator technique** to construct the many-body eigenstates under a proper basis. The complexity of diagonalizing a many-body Hamiltonian using ladder operator techniques only scales linearly with the problem size. Further, from the analytic solutions, we understand more about the systems, such as why certain eigenstates are degenerate. However, I need to point out, there are a few cases that ladder operators get stuck: if more than one group of eigenstates happen to have the same angular momenta, then ladder operators cannot distinguish them. For those cases, we set up the matrix representations in small subspaces and diagonalize small matrices numerically.

The spirit of this chapter is the **first-order degenerate perturbation theory**. Because of the spherical approximation – which makes the Hamiltonian commute with the angular momentum – the $(4l + 2)$ orbitals (including spins) in a shell are energetically degenerate. We set up the many-electron basis in the open-shell, and we solve H_U and H_{SO} in this degenerate basis, which is doing exactly the first-order degenerate perturbation theory. One must realize that the Coulomb Hamiltonian entered already in the DFT calculations: solving H_U again in the open-shell introduces a “double counting” problem. Indeed, the multiplet energies that we are going to obtain are correct up to a constant energy shift. The useful results are the multiplet *splittings*. Further, as we are going to explain, the LS -coupling and jj -coupling schemes introduce yet another first-order degenerate perturbation theory in a different level, where the multiplet terms further split by spin-orbit perturbation and the spin-orbit terms further split by Coulomb perturbation, respectively.

4.2. LS -basis and jj -basis

Consider a group of three electrons. Each has its orbital angular momentum ℓ_i and spin angular momentum $\mathbf{s}_i$.

$$\begin{array}{ccc} \bullet & \bullet & \bullet \\ \ell_1 \mathbf{s}_1 & \ell_2 \mathbf{s}_2 & \ell_3 \mathbf{s}_3 \end{array}$$

Question: What is the total angular momentum of the system?

The answer is simple. But there are two interpretations:

Interpretation 1:

$$\mathbf{J} = (\ell_1 + \ell_2 + \ell_3) + (\mathbf{s}_1 + \mathbf{s}_2 + \mathbf{s}_3) = \sum_{i=1}^3 \ell_i + \sum_{i=1}^3 \mathbf{s}_i = \mathbf{L} + \mathbf{S} \quad (4.5)$$

Interpretation 2:

$$\mathbf{J} = (\boldsymbol{\ell}_1 + \mathbf{s}_1) + (\boldsymbol{\ell}_2 + \mathbf{s}_2) + (\boldsymbol{\ell}_3 + \mathbf{s}_3) = \sum_{i=1}^3 (\boldsymbol{\ell}_i + \mathbf{s}_i) = \sum_{i=1}^3 \mathbf{j}_i \quad (4.6)$$

The two interpretations are equivalent: they are two different perspectives of the same problem.

- In Interpretation 1, the individual orbital angular momenta $\boldsymbol{\ell}_i$ combine together to form a total orbital angular momentum $\mathbf{L}$. The same happens with the individual spin angular momenta $\mathbf{s}_i$ forming a total spin angular momentum $\mathbf{S}$. Then $\mathbf{L}$ and $\mathbf{S}$ sum up together to form the total angular momentum $\mathbf{J}$. This interpretation leads to the *LS*-basis.
- In Interpretation 2, we collect each electron's total angular momentum $\mathbf{j}_i = \boldsymbol{\ell}_i + \mathbf{s}_i$ and then sum them up to get the total angular momentum $\mathbf{J}$ of the system. This interpretation leads to the *jj*-basis.

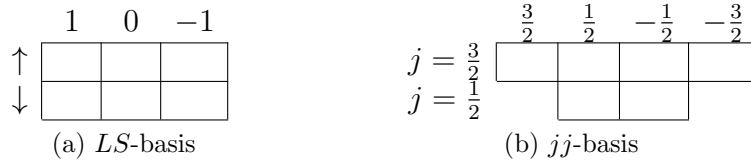


Figure 4.1.: Two different representations of an atomic *p*-shell. (a) A *p*-shell in the *LS*-basis, where an orbital is labelled by orbital projection quantum number m and spin projection quantum number σ ; (b) A *p*-shell in the *jj*-basis, where an orbital is labelled by total angular quantum number j and total angular projection quantum number μ .

Consider a simple atomic *p*-shell system. For an electron in a *p*-shell, it has an orbital angular momentum $l = 1$ and a spin angular momentum (always) $s = 1/2$. A *p*-shell can be filled with up to 6 electrons. Normally, we distinguish the 6 orbitals by the angular projection quantum numbers:

$$m = 1, 0, -1 \quad \text{and} \quad \sigma = \frac{1}{2}, -\frac{1}{2}$$

as shown in Fig. 4.1a. This diagram is in fact in the *LS*-basis, as we distinguish separately $\boldsymbol{\ell}$ and $\mathbf{s}$.

A less familiar representation of a *p*-shell is shown in Fig. 4.1b. In this representation, we sum up $\boldsymbol{\ell}_i$ and $\mathbf{s}_i$ to get $\mathbf{j}_i$. Because electrons are spin $1/2$ particles, unless $l = 0$ (*s*-shell), there will be always two possible values of j , namely,

$$j = l + \frac{1}{2} \quad \text{and} \quad j = l - \frac{1}{2} \quad (4.7)$$

For each j , there is a set of projection quantum numbers. For the *p*-shell, we have

$$\left\{ j = \frac{3}{2}, \mu = \frac{3}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{3}{2} \right\} \quad \text{and} \quad \left\{ j = \frac{1}{2}, \mu = \frac{1}{2}, -\frac{1}{2} \right\} \quad (4.8)$$

4. *LS-coupling and jj-coupling schemes for atomic open-shell problems*

Notice that the number of orbitals is invariant under different coupling schemes. The connection between the *LS*-basis and the *jj*-basis is a unitary transformation (see Section 4.6).

There might be confusion on some notations. I'd better put them more explicit:

1. In the *LS*-basis, a single orbital state is labelled by quantum numbers

$$|n, l, m_l, s, m_s\rangle$$

where m_l and m_s are the projection quantum numbers of l and s , respectively. Here we simplify the notations as $m \leftarrow m_l$ and $\sigma \leftarrow m_s$ to avoid writing the subscripts. Moreover, since s is always $\frac{1}{2}$, it is omitted. Now, we write a single orbital state as

$$|n, l, m, \sigma\rangle$$

where n and l address the shell; m and σ label the different orbitals in the shell.

2. In the *jj*-basis, a single orbital state is labelled by quantum numbers

$$|n, l, s, j, m_j\rangle$$

where j is produced by adding ℓ and $\mathbf{s}$; m_j is the projection quantum number of j . Similarly, m_j is simplified to μ , and $s = \frac{1}{2}$ is omitted. The notation is simplified to

$$|n, l, j, \mu\rangle$$

where n and l address the shell; j and μ label the different orbitals in the shell.

Physics is independent of the choice of basis. So why are we concerned about them? Within a Hilbert space, we have a freedom to choose a basis. But a “good basis” should make the Hamiltonian almost diagonal, therefore easier to obtain the eigen-states. As we are going to explain, when constructing the atomic multiplets, an *LS*-basis makes the Coulomb Hamiltonian H_U almost diagonal, and a *jj*-basis will make the spin-orbit Hamiltonian H_{SO} purely diagonal. Here comes the business:

- The task is to diagonalize a many-body Hamiltonian for an atomic open-shell system.

$$H = H_U + H_{SO}$$

- If H_U dominates (for light atoms like carbon ${}_6\text{C}$ with open-shell $2p^2$), we prefer the “*LS*-basis” to diagonalize H_U exactly and introduce H_{SO} as a perturbation.
- If H_{SO} dominates (rare case but happens for super heavy atoms like flerovium ${}_{114}\text{Fl}$ with open-shell $7p^2$), we prefer the “*jj*-basis” to diagonalize H_{SO} exactly and introduce H_U as a perturbation.

As a remark, in most of the cases, by default, H_{SO} is treated as a perturbation, because the spin-orbit interaction is a relativistic effect, which is normally a weak term. This relativistic effect only grows up for heavy elements. On the other hand, the Coulomb interactions are never that weak. Thus, to have spin-orbit dominated cases, one has to reach super heavy elements like flerovium ${}_{114}\text{Fl}$. We shall confirm this trend systematically in a later chapter.

4.3. The Coulomb and spin-orbit matrix elements

Given a set of basis functions, we can evaluate the corresponding matrix elements. The computation of the Coulomb matrix element $\ddot{U}_{\alpha\beta\gamma\delta}$ is, in general, very difficult. In this chapter, we have simple atomic systems that all the wave functions are centered at the origin (the on-site Coulomb interaction). The evaluation of the on-site $\ddot{U}_{\alpha\beta\gamma\delta}$ is very much easier but still complicated enough that we need to discuss in detail.

The Coulomb matrix element $\ddot{U}_{\alpha\beta\gamma\delta}$ can be expressed as, (see (1.21))

$$\ddot{U}_{\alpha\beta\gamma\delta} = U_{\alpha\beta\gamma\delta} - U_{\alpha\beta\delta\gamma} \quad (4.9)$$

where,

$$\begin{aligned} U_{\alpha\beta\gamma\delta} &= \int d\mathbf{x}_1 \int d\mathbf{x}_2 \varphi_\alpha(\mathbf{x}_1) \varphi_\beta(\mathbf{x}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \varphi_\gamma(\mathbf{x}_2) \varphi_\delta(\mathbf{x}_1) \\ &= \delta_{\sigma_\alpha\sigma_\delta} \delta_{\sigma_\beta\sigma_\gamma} \int d\mathbf{r}_1 \int d\mathbf{r}_2 \varphi_\alpha(\mathbf{r}_1) \varphi_\beta(\mathbf{r}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \varphi_\gamma(\mathbf{r}_2) \varphi_\delta(\mathbf{r}_1) \end{aligned} \quad (4.10)$$

Eqn. (4.10) factorizes to spin and spatial integrals. While the spin part is trivial, the spatial component is rather complicated. The difficulty comes from the term $\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}$, which is 6-dimensional. To integrate over the 6-dimensional function, we use a technique called the “multipole expansion”, which expands the term $\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}$ into a product form of radial and angular parts: [20, 21]

$$\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} = \sum_{k=0}^{\infty} \frac{r_{<}^k}{r_{>}^{k+1}} \frac{4\pi}{2k+1} \sum_{\mu=-k}^k \overline{Y_{k\mu}}(\theta_1, \phi_1) Y_{k\mu}(\theta_2, \phi_2) \quad (4.11)$$

where $r_{<} = \min(r_1, r_2)$ and $r_{>} = \max(r_1, r_2)$. As a result, integral (4.10) divides into radial and angular parts:

$$U_{\alpha\beta\gamma\delta} = \delta_{\sigma_\alpha\sigma_\delta} \delta_{\sigma_\beta\sigma_\gamma} \underbrace{\sum_{k=0}^{\infty} (R_\alpha R_\beta | \frac{r_{<}^k}{r_{>}^{k+1}} | R_\gamma R_\delta)}_{\text{Radial part}} \frac{4\pi}{2k+1} \underbrace{\sum_{\mu=-k}^k \langle Y_\alpha | \overline{Y_{k\mu}} | Y_\delta \rangle \langle Y_\beta | Y_{k\mu} | Y_\gamma \rangle}_{\text{Angular part}} \quad (4.12)$$

The radial part is a Slater integral (or Slater-Condon parameter). This parameter carries the information on the “strength” of the Coulomb interaction: if the wave functions are concentrated (strong Coulomb repulsion), the corresponding Slater integral is large. In Chapter 5, we will discuss the calculation and the periodic trends of this parameter. On the other hand, the angular part consists of two Gaunt coefficients – a Gaunt coefficient is an integral over the product of three spherical harmonics. In Chapter 7, we will discuss the evaluation methods for those special integrals. The multipole expansion (4.11) has an expansion index k that runs from 0 to infinity, which suggests a long summation until convergence. Fortunately, the Gaunt coefficients have nice properties and symmetries so that they vanish under certain combination of the angular momenta. In a simple case,

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

suppose α , β , γ , and δ are orbitals in the same shell (n, l) , the only non-trivial expansion indices are $k = 0, 2, 4, \dots, 2l$.

As a remark, the Coulomb matrix element needs to be evaluated in the LS -basis, because in the LS -basis the space and spin degrees of freedom are separated, thus the explicit integration over the real space is possible.

On the other hand, the spin-orbit matrix elements can be easily evaluated in both the LS -basis and the jj -basis. The spin-orbit matrix element $X_{\alpha\beta}$ has the form:

$$X_{\alpha\beta} = \langle \alpha | \xi(r) \boldsymbol{\ell} \cdot \mathbf{s} | \beta \rangle \quad (4.13)$$

In the LS -basis, the orbital indices α and β consist of the shell numbers (n, l) and the angular momenta (l, m, s, σ) . Thus, $X_{\alpha\beta}$ divides into a radial part and a spin-orbit angular part:

$$X_{\alpha\beta} = \underbrace{\langle R_\alpha | \xi(r) | R_\beta \rangle}_{\text{Radial part}} \underbrace{\langle l_\alpha m_\alpha s_\alpha \sigma_\alpha | \boldsymbol{\ell} \cdot \mathbf{s} | l_\beta m_\beta s_\beta \sigma_\beta \rangle}_{\text{spin-orbit Angular part}} \quad (4.14)$$

The radial part is a simple one-dimensional integral over the radial coordinate, to which we assign a short name – the capital greek $\Xi_{\alpha\beta}$. In Chapter 5, we will discuss the calculation and the periodic trends of this parameter. To work out the angular part, we use the relation:

$$\boldsymbol{\ell} \cdot \mathbf{s} = l_x s_x + l_y s_y + l_z s_z = \frac{1}{2} (l_+ s_- + l_- s_+ + 2l_z s_z) \quad (4.15)$$

Therefore, Eqn. (4.14) becomes,

$$\begin{aligned} X_{\alpha\beta} &= \frac{1}{2} \Xi_{\alpha\beta} \langle l_\alpha m_\alpha s_\alpha \sigma_\alpha | l_+ s_- + l_- s_+ + 2l_z s_z | l_\beta m_\beta s_\beta \sigma_\beta \rangle \\ &= \frac{1}{2} \Xi_{\alpha\beta} \left[\sqrt{(l_\beta + m_\beta + 1)(l_\beta - m_\beta)(\frac{1}{2} + \sigma_\beta)(\frac{1}{2} - \sigma_\beta + 1)} \delta_{m_\alpha, m_\beta + 1} \delta_{\sigma_\alpha, \sigma_\beta - 1} \right. \\ &\quad + \sqrt{(l_\beta + m_\beta)(l_\beta - m_\beta + 1)(\frac{1}{2} + \sigma_\beta + 1)(\frac{1}{2} - \sigma_\beta)} \delta_{m_\alpha, m_\beta - 1} \delta_{\sigma_\alpha, \sigma_\beta + 1} \\ &\quad \left. + 2m_\beta \sigma_\beta \delta_{m_\alpha m_\beta} \delta_{\sigma_\alpha \sigma_\beta} \right] \end{aligned} \quad (4.16)$$

which is the spin-orbit matrix element evaluated in the LS -basis.

In the jj -basis, the orbital indices α and β consist of the shell numbers (n, l) and the angular momenta (l, s, j, μ) . Thus,

$$X_{\alpha\beta} = \underbrace{\langle R_\alpha | \xi(r) | R_\beta \rangle}_{\text{Radial part}} \underbrace{\langle l_\alpha s_\alpha j_\alpha \mu_\alpha | \boldsymbol{\ell} \cdot \mathbf{s} | l_\beta s_\beta j_\beta \mu_\beta \rangle}_{\text{spin-orbit Angular part}} \quad (4.17)$$

The radial part is the same $\Xi_{\alpha\beta}$ as evaluated in the LS -basis (4.14). To work out the angular part, we use the relation:²

$$\boldsymbol{\ell} \cdot \mathbf{s} = \frac{1}{2} (\hat{j}^2 - \hat{l}^2 - \hat{s}^2) \quad (4.18)$$

²Here we add “hats” on the operators to distinguish them from the quantum numbers, because in this context there exists an ambiguity without the “hats”. However, always carrying the “hats” on operators is bit cumbersome. Normally, when there is no ambiguity, the “hats” are dropped.

Now, Eqn. (4.17) becomes,

$$\begin{aligned} X_{\alpha\beta} &= \frac{1}{2} \Xi_{\alpha\beta} \langle l_{\alpha} s_{\alpha} j_{\alpha} \mu_{\alpha} | \hat{j}^2 - \hat{l}^2 - \hat{s}^2 | l_{\beta} s_{\beta} j_{\beta} \mu_{\beta} \rangle \\ &= \frac{1}{2} \Xi_{\alpha\beta} [j_{\beta}(j_{\beta} + 1) - l_{\beta}(l_{\beta} + 1) - \frac{3}{4}] \delta_{l_{\alpha} l_{\beta}} \delta_{j_{\alpha} j_{\beta}} \delta_{\mu_{\alpha} \mu_{\beta}} \end{aligned} \quad (4.19)$$

which is the spin-orbit matrix element evaluated in the jj -basis.

4.4. The LS -coupling scheme

Solving the open-shell problems in the LS -coupling scheme yields the familiar *atomic multiplets* [15]. In this section, we summarize the main ideas of the LS -coupling scheme.

The problem is to diagonalize

$$H = H_U + H_{SO} \quad (4.20)$$

in an atomic shell. In the LS -coupling scheme, we first diagonalize H_U exactly, and then solve H_{SO} as a perturbation.

4.4.1. Diagonalizing H_U in the LS -basis

Choosing the LS -basis, one can take the advantage of the symmetry that the Coulomb Hamiltonian H_U commutes with the total angular momenta $\mathbf{S}$ and $\mathbf{L}$.

Proof:

We want to prove that

$$[H_U, \mathbf{S}] = 0 \quad \text{and} \quad [H_U, \mathbf{L}] = 0 \quad (4.21)$$

where,

$$H_U = \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad \mathbf{S} = \sum_{i=1}^{N_e} \mathbf{s}_i, \quad \mathbf{L} = \sum_{i=1}^{N_e} \mathbf{l}_i \quad (4.22)$$

$[H_U, \mathbf{S}] = 0$ is trivial, since H_U has no spin dependence. The physical picture is that flipping spins has no effect on the Coulomb interaction.

$[H_U, \mathbf{L}] = 0$ is a bit more difficult. We need

$$\mathbf{l}_i = -i \mathbf{r}_i \times \nabla_i \quad (4.23)$$

Consider the commutator of a Coulomb pair and an angular momentum operator, (working with differential operators, we need a “test function” $f(\mathbf{r}_i)$, so that we don’t

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

forget the product rule)

$$\begin{aligned}
 \left[\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \ell_i \right] f(\mathbf{r}_i) &= \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \ell_i f(\mathbf{r}_i) - \ell_i \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} f(\mathbf{r}_i) \\
 &= \frac{-i\mathbf{r}_i}{|\mathbf{r}_i - \mathbf{r}_j|} \times \nabla_i f(\mathbf{r}_i) + i\mathbf{r}_i \times \underbrace{\nabla_i \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}}_{\text{product rule}} f(\mathbf{r}_i) \\
 &= \frac{-i\mathbf{r}_i}{|\mathbf{r}_i - \mathbf{r}_j|} \times \nabla_i f(\mathbf{r}_i) - i\mathbf{r}_i \times \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} f(\mathbf{r}_i) + \frac{i\mathbf{r}_i}{|\mathbf{r}_i - \mathbf{r}_j|} \times \nabla_i f(\mathbf{r}_i)
 \end{aligned}$$

The first and the last terms cancel. In the middle term, there are $\mathbf{r}_i \times \mathbf{r}_i = 0$ and $\mathbf{r}_i \times \mathbf{r}_j = -\mathbf{r}_j \times \mathbf{r}_i$. Dropping the test function, we have,

$$\left[\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \ell_i \right] = \frac{i\mathbf{r}_i \times \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} = -\frac{i\mathbf{r}_j \times \mathbf{r}_i}{|\mathbf{r}_i - \mathbf{r}_j|^3} = -\left[\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \ell_j \right] \quad (4.24)$$

Thus, a Coulomb pair doesn't commute with a single angular momentum operator, but it commutes with a pair of angular momenta:

$$\left[\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \ell_i + \ell_j \right] = 0 \quad \Rightarrow \quad \left[\sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \sum_{i=1}^{N_e} \ell_i \right] = 0 \quad \text{Q.E.D.} \quad (4.25)$$

The physical picture is that the Coulomb interaction is invariant under the rotation of the entire system: If we pick one electron and rotate it around the origin, the system is certainly altered; If we rotate a pair of electrons, the Coulomb interaction between the pair stays the same (that's why they commute pair-wise); If we rotate all the electrons, the Coulomb interactions among all the pairs are unchanged.

Given that $[H_U, \mathbf{S}] = 0$ and $[H_U, \mathbf{L}] = 0$, we have the following relations:

$$[H_U, S^2] = 0 \quad [H_U, L^2] = 0 \quad [H_U, S_z] = 0 \quad [H_U, L_z] = 0 \quad (4.26)$$

As a result, an eigen-state of H_U is simultaneously eigen-state of S^2, L^2, S_z, L_z . Besides, the four operators *mutually commute*. Thus, we can label an eigen-state of H_U with the four quantum numbers: S, L, M_S, M_L .

A side note, because $\mathbf{J} = \mathbf{L} + \mathbf{S}$, it follows that $[H_U, \mathbf{J}] = 0$, and consequently $[H_U, J^2] = [H_U, J_z] = 0$. An eigen-state of H_U can be also labelled by two quantum numbers: J, M_J . However, it can't be simultaneously labelled by S, L, J, M_S, M_L, M_J (that would be cool), because the operators $S^2, L^2, J^2, S_z, L_z, J_z$ do not *mutually commute* (e.g. $[J^2, L_z] \neq 0$).

The commutation relations $[H_U, \mathbf{S}] = 0$ and $[H_U, \mathbf{L}] = 0$ also lead to

$$[H_U, S_{\pm}] = 0 \quad \text{and} \quad [H_U, L_{\pm}] = 0 \quad (4.27)$$

The logical consequence is:

If: $\Psi\rangle$ is an eigen-state of H_U with eigen-energy E, Then: $S_{\pm} \Psi\rangle$ and $L_{\pm} \Psi\rangle$ are also eigen-states of H_U with the same eigen-energy E.
--

4.4. The LS -coupling scheme

The ladder operators only change the projection quantum numbers M_S and M_L . Thus, the eigen-states that differ only in M_S and M_L are degenerate. The states that have the same S and L form a group called the *multiplet term*.³ A multiplet term has the symbol

$$^{2S+1}L \quad (4.28)$$

We denote each eigen-state as

$$|^{2S+1}L_{M_S M_L}\rangle \quad (4.29)$$

Within a multiplet term,

$$\begin{aligned} M_S &= -S, -S+1, \dots, -1, 0, 1, \dots, S-1, S \\ M_L &= -L, -L+1, \dots, -1, 0, 1, \dots, L-1, L \end{aligned}$$

The degeneracy of a multiplet term is $(2S+1)(2L+1)$.

Here we demonstrate solving the p^2 system, which is the simplest non-trivial system to study. We set up the many-electron basis of the p^2 system, as shown in Fig. 4.2.

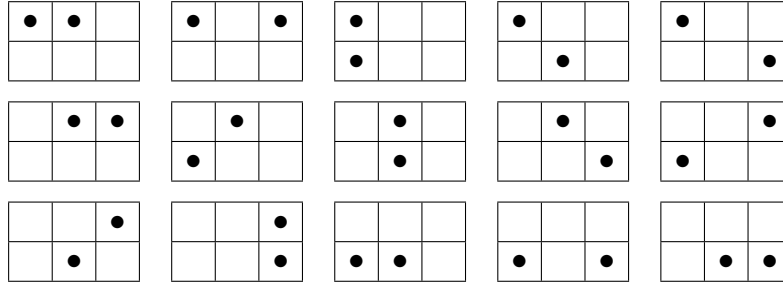


Figure 4.2.: The many-electron basis states of a p^2 shell in the LS -coupling scheme. The orbital quantum numbers are given in Fig. 4.1a.

Each basis state has well defined

$$M_S = \sum_{i=1}^{N_e} \sigma_i \quad \text{and} \quad M_L = \sum_{i=1}^{N_e} m_i \quad (4.30)$$

In general,

An eigen-state $|^{2S+1}L_{M_S M_L}\rangle$ is a linear combination of the basis states that have the same (M_S, M_L) .

For all basis states, we mark their (M_S, M_L) . The corresponding values in Fig. 4.2 are

$$\begin{pmatrix} 1, & 1 \end{pmatrix} \begin{pmatrix} 1, & 0 \end{pmatrix} \begin{pmatrix} 0, & 2 \end{pmatrix} \begin{pmatrix} 0, & 1 \end{pmatrix} \begin{pmatrix} 0, & 0 \end{pmatrix} \\ \begin{pmatrix} 1, & -1 \end{pmatrix} \begin{pmatrix} 0, & 1 \end{pmatrix} \begin{pmatrix} 0, & 0 \end{pmatrix} \begin{pmatrix} 0, & -1 \end{pmatrix} \begin{pmatrix} 0, & 0 \end{pmatrix} \\ \begin{pmatrix} 0, & -1 \end{pmatrix} \begin{pmatrix} 0, & -2 \end{pmatrix} \begin{pmatrix} -1, & 1 \end{pmatrix} \begin{pmatrix} -1, & 0 \end{pmatrix} \begin{pmatrix} -1, & -1 \end{pmatrix}$$

³It is possible that two multiplet terms happen to have the same S and L .

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

We can quickly obtain the multiplet terms of a system by a simple “counting”. We count the *number* of basis states with the same (M_S, M_L) . For the p^2 system, we have:

		M_L				
		2	1	0	-1	-2
M_S	1	0	1	1	1	0
	0	1	2	3	2	1
	-1	0	1	1	1	0

Now, we collect groups, in which the group members are connected by the ladder operators $L_{\pm}$ and $S_{\pm}$:

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{cases} \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} & (S = 1, L = 1) \rightarrow {}^3P \\ + \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & (S = 0, L = 2) \rightarrow {}^1D \\ + \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & (S = 0, L = 0) \rightarrow {}^1S \end{cases}$$

The corresponding eigen-states $|^{2S+1}L_{M_S M_L}\rangle$ are:

$ ^3P_{11}\rangle$	$ ^3P_{10}\rangle$	$ ^3P_{1-1}\rangle$		
$ ^3P_{01}\rangle$	$ ^3P_{00}\rangle$	$ ^3P_{0-1}\rangle$		
$ ^3P_{-11}\rangle$	$ ^3P_{-10}\rangle$	$ ^3P_{-1-1}\rangle$		
$ ^1D_{02}\rangle$	$ ^1D_{01}\rangle$	$ ^1D_{00}\rangle$	$ ^1D_{0-1}\rangle$	$ ^1D_{0-2}\rangle$
$ ^1S_{00}\rangle$				

By counting the basis states, we can easily produce the *names* of the eigen-states. But we haven’t really *solved* for them, which means to express the eigen-states in terms of the basis states. Here enters the ladder operator technique, and we briefly discuss how this method works:

Within a multiplet group, there are $(2S + 1)(2L + 1)$ eigen-states. If we know (any) one of the eigen-states, we can construct the entire group using the ladder operators. Among the basis states (Fig. 4.2), some are – by themselves – eigen-states. Those are the ones that have a unique (M_S, M_L) . Referring back to the M_S - M_L table of the p^2 problem,

$$\begin{pmatrix} 0 & \boxed{1} & \boxed{1} & \boxed{1} & 0 \\ \boxed{1} & 2 & 3 & 2 & \boxed{1} \\ 0 & \boxed{1} & \boxed{1} & \boxed{1} & 0 \end{pmatrix}$$

4.4. The LS -coupling scheme

All the basis states (marked by boxes) that have a unique (M_S, M_L) can serve as a starting eigen-state. For instance,

$$\begin{pmatrix} 0 & \boxed{1} & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \mapsto \begin{array}{|c|c|c|} \hline \bullet & \bullet & \\ \hline & & \\ \hline & & \\ \hline \end{array} \mapsto c_{0\uparrow}^\dagger c_{1\uparrow}^\dagger |0\rangle \quad (4.31)$$

This basis state has a unique $(M_S, M_L) = (1, 1)$. It must be the eigen-state

$$|^3P_{11}\rangle = c_{0\uparrow}^\dagger c_{1\uparrow}^\dagger |0\rangle \quad (4.32)$$

Now we can apply ladder operators to get the entire 3P group. Recall the “ladder relations”,

$$L_\pm |l, m\rangle = \alpha_{lm}^\pm |l, m \pm 1\rangle \quad \text{where} \quad \alpha_{lm}^\pm = \sqrt{(l \mp m)(l \pm m + 1)} \quad (4.33)$$

For instance,

$$L_- |^3P_{11}\rangle = \sqrt{2} |^3P_{10}\rangle \quad (4.34)$$

On the other hand,

$$\begin{aligned} L_- c_{0\uparrow}^\dagger c_{1\uparrow}^\dagger |0\rangle &= (\ell_- c_{0\uparrow}^\dagger) c_{1\uparrow}^\dagger |0\rangle + c_{0\uparrow}^\dagger (\ell_- c_{1\uparrow}^\dagger) |0\rangle \\ &= \sqrt{2} c_{-1\uparrow}^\dagger c_{1\uparrow}^\dagger |0\rangle + \underbrace{\sqrt{2} c_{0\uparrow}^\dagger c_{0\uparrow}^\dagger}_{\text{Pauli} \rightarrow 0} |0\rangle \end{aligned} \quad (4.35)$$

Equating (4.34) and (4.35), we see that,

$$|^3P_{10}\rangle = c_{-1\uparrow}^\dagger c_{1\uparrow}^\dagger |0\rangle \quad (4.36)$$

Oh, but this is not very interesting: it is just another basis state. What happened is that we used L_- to produce $|^3P_{10}\rangle$:

$$\begin{pmatrix} 0 & \boxed{1} & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \xrightarrow{L_-} \begin{pmatrix} 0 & 1 & \boxed{1} & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

It is not necessary to use ladder operators to produce this state (because we know already it is a single basis state), but we *can* use ladder operators to produce it, and the results are consistent.

How about to try S_- :

$$S_- |^3P_{11}\rangle = \sqrt{2} |^3P_{01}\rangle \quad (4.37)$$

On the other hand,

$$\begin{aligned} S_- c_{0\uparrow}^\dagger c_{1\uparrow}^\dagger |0\rangle &= (s_- c_{0\uparrow}^\dagger) c_{1\uparrow}^\dagger |0\rangle + c_{0\uparrow}^\dagger (s_- c_{1\uparrow}^\dagger) |0\rangle \\ &= c_{0\downarrow}^\dagger c_{1\uparrow}^\dagger |0\rangle + c_{0\uparrow}^\dagger c_{1\downarrow}^\dagger |0\rangle \\ &= c_{0\downarrow}^\dagger c_{1\uparrow}^\dagger |0\rangle - c_{1\downarrow}^\dagger c_{0\uparrow}^\dagger |0\rangle \quad (\text{ordering}) \end{aligned} \quad (4.38)$$

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

The last step in (4.38) is to keep the default orbital ordering (phase convention). Therefore,

$$|^3P_{01}\rangle = \frac{1}{\sqrt{2}} \left(c_{0\downarrow}^\dagger c_{1\uparrow}^\dagger - c_{1\downarrow}^\dagger c_{0\uparrow}^\dagger \right) |0\rangle \quad (4.39)$$

We used S_- to produce $^3P_{01}$:

$$\begin{pmatrix} 0 & \boxed{1} & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \xrightarrow{S_-} \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & \boxed{1} & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

We can keep applying S_- and L_- in the same manner to obtain the entire 9 eigen-states. They are:

$$\begin{aligned} |^3P_{11}\rangle &= c_{0\uparrow}^\dagger c_{1\uparrow}^\dagger |0\rangle \\ |^3P_{01}\rangle &= \frac{1}{\sqrt{2}} \left(c_{0\downarrow}^\dagger c_{1\uparrow}^\dagger - c_{1\downarrow}^\dagger c_{0\uparrow}^\dagger \right) |0\rangle \\ |^3P_{-11}\rangle &= c_{0\downarrow}^\dagger c_{1\downarrow}^\dagger |0\rangle \\ |^3P_{10}\rangle &= c_{-1\uparrow}^\dagger c_{1\uparrow}^\dagger |0\rangle \\ |^3P_{00}\rangle &= \frac{1}{\sqrt{2}} \left(c_{-1\downarrow}^\dagger c_{1\uparrow}^\dagger - c_{1\downarrow}^\dagger c_{-1\uparrow}^\dagger \right) |0\rangle \\ |^3P_{-10}\rangle &= c_{-1\downarrow}^\dagger c_{1\downarrow}^\dagger |0\rangle \\ |^3P_{1-1}\rangle &= c_{-1\uparrow}^\dagger c_{0\uparrow}^\dagger |0\rangle \\ |^3P_{0-1}\rangle &= \frac{1}{\sqrt{2}} \left(c_{-1\downarrow}^\dagger c_{0\uparrow}^\dagger - c_{0\downarrow}^\dagger c_{-1\uparrow}^\dagger \right) |0\rangle \\ |^3P_{-1-1}\rangle &= c_{-1\downarrow}^\dagger c_{0\downarrow}^\dagger |0\rangle \end{aligned}$$

That completes the (9-fold degenerate) 3P group.

Likewise, we can choose to start from the basis state with the unique $(M_S, M_L) = (0, 2)$, which corresponds to the eigen-state $^1D_{02}$:

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ \boxed{1} & 2 & 3 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \mapsto \left| \begin{array}{|c|c|c|} \hline \bullet & & \\ \hline \bullet & & \\ \hline \end{array} \right\rangle \mapsto c_{1\downarrow}^\dagger c_{1\uparrow}^\dagger |0\rangle \quad (4.40)$$

Starting from this state, we can apply L_- (4 times), to construct the entire 1D group.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The (5-fold degenerate) 1D eigen-states are:

$$\begin{aligned} |^1D_{0-2}\rangle &= c_{1\downarrow}^\dagger c_{1\uparrow}^\dagger |0\rangle \\ |^1D_{0-1}\rangle &= \frac{1}{\sqrt{2}} \left(c_{0\downarrow}^\dagger c_{1\uparrow}^\dagger + c_{1\downarrow}^\dagger c_{0\uparrow}^\dagger \right) |0\rangle \\ |^1D_{0-0}\rangle &= \frac{1}{\sqrt{6}} \left(c_{-1\downarrow}^\dagger c_{1\uparrow}^\dagger + 2c_{0\downarrow}^\dagger c_{0\uparrow}^\dagger + c_{1\downarrow}^\dagger c_{-1\uparrow}^\dagger \right) |0\rangle \\ |^1D_{0-1}\rangle &= \frac{1}{\sqrt{2}} \left(c_{-1\downarrow}^\dagger c_{0\uparrow}^\dagger + c_{0\downarrow}^\dagger c_{-1\uparrow}^\dagger \right) |0\rangle \\ |^1D_{0-2}\rangle &= c_{-1\downarrow}^\dagger c_{-1\uparrow}^\dagger |0\rangle \end{aligned}$$

We have constructed the eigen-states for 3P and 1D . The remaining job is to figure out 1S , in which there is only one eigen-state $|^1S_{00}\rangle$. If you think about it, you will realize that we cannot start from any basis state to apply ladder operators. This time, we need to use the basis orthogonalization relation. The idea is the following:

The three states – $|^3P_{00}\rangle$, $|^1D_{00}\rangle$, $|^1S_{00}\rangle$ – have the same $(M_S, M_L) = (0, 0)$. They all live in the (3-dim) space spanned by $\{c_{-1\downarrow}^\dagger c_{1\uparrow}^\dagger |0\rangle, c_{0\downarrow}^\dagger c_{0\uparrow}^\dagger |0\rangle, c_{1\downarrow}^\dagger c_{-1\uparrow}^\dagger |0\rangle\}$, which are the basis states with $(M_S, M_L) = (0, 0)$. Now, $|^3P_{00}\rangle$ and $|^1D_{00}\rangle$ are known, $|^1S_{00}\rangle$ is an unknown:

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & \boxed{3} & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{cases} \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & \boxed{1} & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \mapsto |^3P_{00}\rangle \mapsto \mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \\ + \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & \boxed{1} & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mapsto |^1D_{00}\rangle \mapsto \mathbf{v}_2 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \\ + \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mapsto |^1S_{00}\rangle \mapsto \mathbf{v}_3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \end{cases}$$

Given that $\langle \mathbf{v}_3 | \mathbf{v}_1 \rangle = 0$, $\langle \mathbf{v}_3 | \mathbf{v}_2 \rangle = 0$, and $\langle \mathbf{v}_3 | \mathbf{v}_3 \rangle = 1$, we can solve for $\mathbf{v}_3$. This is a typical “basis completion” problem. Here $\mathbf{v}_3$ can be somehow figured out easily: it is a normalized $[1, -1, 1]$. In general, such a problem can be solved systematically by singular value decomposition. In this case,

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix} = \mathbf{U} \mathbf{S} \mathbf{V}^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix}$$

The last row in $\mathbf{V}^\dagger$ is the solution vector that we are looking for. The result is:

$$|^1S_{00}\rangle = \frac{1}{\sqrt{3}} \left(c_{-1\downarrow}^\dagger c_{1\uparrow}^\dagger - c_{0\downarrow}^\dagger c_{0\uparrow}^\dagger + c_{1\downarrow}^\dagger c_{-1\uparrow}^\dagger \right) |0\rangle$$

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

Therefore, we conclude that we have constructed the (9-fold degenerate) 3P , (5-fold degenerate) 1D , and (1-fold degenerate) 1S eigen-states, using ladder operators and orthogonalization relations. That is the technique for (analytically) diagonalizing the Coulomb Hamiltonian H_U .

Our discussion in this chapter focuses on constructing the many-electron *eigen-states*. The most important physical quantities out of the states are, after all, the *eigen-energies*. The multiplet term energies can be *analytically* expressed in terms of the Slater-Condon parameters and the Gaunt coefficients (4.12). For the explicit eigen-energy expressions, see Appendix B.

For general open-shell systems, we set up the many-electron basis using Algorithm 4.1. The BasisLS class inherits from the general Basis class. It additionally assigns the individual orbitals and the many-electron states physical meanings, namely, the quantum numbers. It can apply ladder operators to a vector in the many-electron basis. Algorithm 4.2 produces multiplet terms for open-shell systems with the shell number l and the number of electrons N_e .

Algorithm 4.1: The BasisLS class (inherit from Basis (Algorithm 1.1)).

```

1 class BasisLS(Basis):
2     def __init__(self, l, Ne):
3         Basis.__init__(self, 4*l+2, Ne)
4         self.l = l # Shell l
5
6         # Quantum numbers of individual orbitals
7         self.ml = [1-a*(2*l+1) for a in range(self.Norb)] # Orbital ml
8         self.ms = [0.5-a/(2*l+1) for a in range(self.Norb)] # Orbital ms
9
10        # Quantum numbers of many-electron basis
11        self.ML = [self.getML(iconf) for iconf in self.conf] # Basis total ML
12        self.MS = [self.getMS(iconf) for iconf in self.conf] # Basis total MS
13
14        # Calculate total ML of iconf
15        def getML(self, iconf):
16            return sum(self.ml[a] for a in range(self.Norb) if isBit(iconf, a))
17
18        # Calculate total MS of iconf
19        def getMS(self, iconf):
20            return sum(self.ms[a] for a in range(self.Norb) if isBit(iconf, a))
21
22        # Apply L- to a vector (in the many-electron basis)
23        def Lm(self, v):
24            w = np.zeros(self.dim)
25            # Apply L- to non-zero basis states
26            for i in np.nonzero(v)[0]:
27                iconf = self.conf[i]
28                # Orbitals with ml > -l
29                for a in list(range(self.Norb//2-1))+list(range(self.Norb//2, self.Norb-1)):
30                    # Apply l-
31                    b = a + 1
32                    if isBit(iconf, a) and not isBit(iconf, b):
33                        jconf = setBit(clearBit(iconf, a), b)
34                        j = self.index[jconf]
35                        alpha = math.sqrt((self.l+self.ml[a])*(self.l-self.ml[a]+1))
36                        w[j] += alpha*v[i] # fsign = 1
37            return w/np.linalg.norm(w)
38
39        # Apply S- to a vector (in the many-electron basis)

```

```

40 def Sm(self, v):
41     w = np.zeros(self.dim)
42     # Apply S- to non-zero basis states
43     for i in np.nonzero(v)[0]:
44         iconf = self.conf[i]
45         # Orbitals with up-spin
46         for a in range(self.Norb//2):
47             # Apply s-
48             b = a + self.Norb//2
49             if isBit(iconf, a) and not isBit(iconf, b):
50                 jconf = setBit(clearBit(iconf, a), b)
51                 j = self.index[jconf]
52                 mask = (-1<(a+1))^(1<b)
53                 fsign = 1-2*(countBit(iconf&mask)&1)
54                 w[j] += fsign*v[i] # alpha = 1
55     return w/np.linalg.norm(w)

```

Algorithm 4.2: Construct multiplet terms for an open-shell system

```

1  from basis import BasisLS
2  import numpy as np
3
4  def multiplet_terms(l, Ne):
5      # Set up basis
6      B = BasisLS(l, Ne)
7
8      # Set up MS-ML table (MS can be half-integer)
9      MSList = sorted(set(B.MS), reverse=True)
10     MLList = sorted(set(B.ML), reverse=True)
11     MSML = list(zip(B.MS, B.ML)) # Basis (MS, ML) tuples
12     table = {MS: {ML: MSML.count((MS,ML)) for ML in MLList} for MS in MSList}
13
14     # Scan the table and collect multiplet terms
15     terms = []
16     for S in MSList[:len(MSList)//2+1]:
17         for L in MLList[:len(MLList)//2+1]:
18             N = table[S][L]
19             if N>0:
20                 for MS in np.arange(-S, S+1):
21                     for ML in np.arange(-L, L+1):
22                         table[MS][ML] -= N
23                         terms.append((S, L, N))
24
25     # Construct multiplet eigen-vectors
26     vecs = {}
27     for (S, L, N) in terms:
28         for sn in range(N):
29             vecs[(S,L,sn)] = {MS: {ML: np.zeros(B.dim) for ML in np.arange(-L, L+1)}
30                               for MS in np.arange(-S, S+1)}
31
32     # Prepare (N-number-of) leading vectors (leading means MS=S, ML=L)
33     if N==1 and MSML.count((S,L))==1:
34         # Unique basis state
35         vecs[(S,L,0)][S][L][MSML.index((S,L))] = 1
36     else:
37         # Previous vecs with MS=S, ML=L
38         P = [pv[S][L] for ((pS,pL,psn),pv) in vecs.items()
39               if (pS>=S and pL>L) or (pS>S and pL>=L)]
40         # Solve an SVD problem to complete the orthonormal basis
41         nz = [MSMLi==(S,L) for MSMLi in MSML]
42         A = [p[nz] for p in P]
43         vh = np.linalg.svd(A)[2]
44         for sn in range(N):
45             vecs[(S,L,sn)][S][L][nz] = vh[len(P)+sn]
46

```

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

```

47     # Ladder operator technique
48     for sn in range(N):
49         for MS in np.arange(S, -S, -1):
50             vecs[(S,L,sn)][MS-1][L] = B.Sm(vecs[(S,L,sn)][MS][L]) # Apply S-
51         for MS in np.arange(S, -S-1, -1):
52             for ML in np.arange(L, -L, -1):
53                 vecs[(S,L,sn)][MS][ML-1] = B.Lm(vecs[(S,L,sn)][MS][ML]) # Apply L-
54     return (terms, vecs, B)

```

The function `multiplet_terms` returns a tuple of results: `terms` is a list of multiplet names. Each term in `terms` is a tuple (S, L, N) , which represents the multiplet name: $N \times {}^{2S+1}L$, where N is the number of seniorities (will be explained shortly). `vecs` is the collection of the (eigen-)vectors (constructed by the ladder operators) in the basis B . In general, `vecs` is a huge object. It contains $\dim_{\mathbb{H}}$ vectors, and each vector contains $\dim_{\mathbb{H}}$ elements. However, many of the elements in the vectors are zeros, because the LS -basis is a “good basis” for the Coulomb Hamiltonian. (Note: to print out `vecs`, it’s better to output only the *non-zero* elements and the corresponding basis states. For example, I would prefer to output a string in the TeX format like (4.32). (I wouldn’t leave the code here because it is highly customizable and can look ugly))

Some examples: (here we do not print out the `vecs`, that will consume too much space)

```
>>> multiplet_terms(1, 2)[0]
```

```
[(1,1,1), (0,2,1), (0,0,1)]
```

That is the p^2 system:

$${}^3P, {}^1D, {}^1S$$

```
>>> multiplet_terms(2, 3)[0]
```

```
[(1.5,3,1), (1.5,1,1), (0.5,5,1), (0.5,4,1), (0.5,3,1), (0.5,2,2), (0.5,1,1)]
```

That is a d^3 system:

$${}^4F, {}^4P, {}^2H, {}^2G, {}^2F, 2 \times {}^2D, {}^2P$$

In a d^3 system, there are two 2D terms. They appear to have the same name, but they are actually two different groups, with different eigen-energies. In the general situation, ${}^{2S+1}L$ cannot address a group uniquely. The “seniority number” was introduced by Giulio Racah [22] to distinguish the groups with the same total S and total L . But physically, there is nothing new (or deep) here: just two groups happen to have the same total angular momenta. You might insist to ask, “Well, if the two groups happen to have the same angular momenta, why don’t they just ‘merge’ into one group?” Remember, for a given multiplet term, there are precisely $(2S+1)(2L+1)$ group members. So a term has a limited “number of seats”. But the amount of basis states is often enormous, in the way that certain (M_S, M_L) ’s appear too often, which, by construction, enforces certain multiplet terms occur more than once.

The “seniority numbers” lacks physical meanings. For a given $N \times {}^{2S+1}L$, we can simply

denote the groups as⁴

$$^{2S+1}L^{\text{sn}} \quad \text{where, } \text{sn} = 0, 1, 2, \dots, N-1 \quad (4.41)$$

Correspondingly, in Algorithm 4.2, a state $|^{2S+1}L_{M_S M_L}^{\text{sn}}\rangle$ is represented by

$$\text{vecs}[(S, L, \text{sn})] [\text{MS}] [\text{ML}]$$

The “seniority numbers” do not introduce a new physical meaning, but they do introduce *troubles*. For instance, within the $2 \times 2D$ groups, the states that we obtained from `vecs` in the routine above are actually *not* eigen-states. The reason is the following: When we use the ladder operator technique, we need to start from *one* leading eigen-state (obtained from either a unique basis state or by completing the basis). However, for the $2 \times 2D$ case, we have to start from *two* leading eigen-states. By completing the basis (here we use the `numpy.linalg.svd`), we get two new states, they span a 2-dim space. Notice that *any* two states in this 2-dim space are orthogonal to the previous eigen-states. The two leading eigen-states live inside this 2-dim plane, but in general, we cannot determine them. This is the situation when the ladder operator technique gets stuck and we cannot diagonalize H_U fully analytical (what a pity). In general, for an $N \times ^{2S+1}L$ case, N leading eigen-states have to be determined numerically, because they depend on the values of the matrix elements. Luckily, we only need to set up a small matrix representation in the N -dimensional subspace, and solve a small N -by- N matrix problem numerically. Once the N leading eigen-states are determined, we can continue applying the ladder operators to get the rests.

4.4.2. Diagonalizing H_{SO} in the LS -basis

We have completed the discussion of diagonalizing H_U analytically in the LS -basis. Now we ask: what if we introduce the spin-orbit Hamiltonian H_{SO} ? The problem is that the spin-orbit Hamiltonian H_{SO} does not commute with the operators $\mathbf{S}$ or $\mathbf{L}$. Consequently, S, L, M_S, M_L are not the proper quantum numbers for the eigen-states of H_{SO} . However, H_{SO} does commute with the total angular momentum $\mathbf{J}$.

Proof:

We want to prove that

$$[H_{\text{SO}}, \mathbf{L}] \neq 0, \quad [H_{\text{SO}}, \mathbf{S}] \neq 0, \quad [H_{\text{SO}}, \mathbf{J}] = 0 \quad (4.42)$$

where

$$H_{\text{SO}} = \sum_{i=1}^{N_e} \xi(r_i) \boldsymbol{\ell}_i \cdot \mathbf{s}_i \quad \text{and} \quad \mathbf{J} = \mathbf{L} + \mathbf{S} \quad (4.43)$$

Consider the individual components $[\boldsymbol{\ell}_i \cdot \mathbf{s}_i, \boldsymbol{\ell}_i]$ and $[\boldsymbol{\ell}_i \cdot \mathbf{s}_i, \mathbf{s}_i]$: (here we drop index i

⁴This is not the original definition of the seniority number [22]. Here we simplified the discussion.

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

for cleanness)

$$\begin{aligned} [\boldsymbol{\ell} \cdot \mathbf{s}, \boldsymbol{\ell}] &= [\ell_x \hat{\mathbf{x}} + \ell_y \hat{\mathbf{y}} + \ell_z \hat{\mathbf{z}}] \\ &= [\ell_x \hat{\mathbf{x}}, \boldsymbol{\ell}] + [\ell_y \hat{\mathbf{y}}, \boldsymbol{\ell}] + [\ell_z \hat{\mathbf{z}}, \boldsymbol{\ell}] \end{aligned}$$

Focusing on the first term:

$$\begin{aligned} [\ell_x \hat{\mathbf{x}}, \boldsymbol{\ell}] &= [\ell_x s_x, \ell_x \hat{\mathbf{x}}] + [\ell_y s_y, \ell_x \hat{\mathbf{x}}] + [\ell_z s_z, \ell_x \hat{\mathbf{x}}] \\ &= s_x [\ell_x, \ell_x] \hat{\mathbf{x}} + s_y [\ell_y, \ell_x] \hat{\mathbf{x}} + s_z [\ell_z, \ell_x] \hat{\mathbf{x}} \\ &= s_x(0) \hat{\mathbf{x}} + s_y(-i\ell_z) \hat{\mathbf{x}} + s_z(i\ell_y) \hat{\mathbf{x}} \\ &= i(\ell_y s_z - \ell_z s_y) \hat{\mathbf{x}} \\ &= i(\boldsymbol{\ell} \times \mathbf{s})_x \hat{\mathbf{x}} \end{aligned}$$

The same argument applies to the other two components. Therefore,

$$[\boldsymbol{\ell} \cdot \mathbf{s}, \boldsymbol{\ell}] = i(\boldsymbol{\ell} \times \mathbf{s})_x \hat{\mathbf{x}} + i(\boldsymbol{\ell} \times \mathbf{s})_y \hat{\mathbf{y}} + i(\boldsymbol{\ell} \times \mathbf{s})_z \hat{\mathbf{z}} = i(\boldsymbol{\ell} \times \mathbf{s}) \quad (4.44)$$

Likewise,

$$[\boldsymbol{\ell} \cdot \mathbf{s}, \mathbf{s}] = i(\mathbf{s} \times \boldsymbol{\ell}) \quad (4.45)$$

Therefore, the spin-orbit Hamiltonian does not commute with total $\mathbf{L}$ or total $\mathbf{S}$,

$$[H_{\text{SO}}, \mathbf{L}] = i \sum_{i=1}^{N_e} \xi(\mathbf{r}_i) \boldsymbol{\ell}_i \times \mathbf{s}_i \quad [H_{\text{SO}}, \mathbf{S}] = i \sum_{i=1}^{N_e} \xi(\mathbf{r}_i) \mathbf{s}_i \times \boldsymbol{\ell}_i \quad (4.46)$$

However, it commutes with the total $\mathbf{J}$, (the two above cancel each other)

$$[H_{\text{SO}}, \mathbf{J}] = [H_{\text{SO}}, \mathbf{L}] + [H_{\text{SO}}, \mathbf{S}] = 0 \quad \text{Q.E.D.} \quad (4.47)$$

The physical intuition is that the spin-orbit interaction is invariant if one rotates all degrees of freedom. Now, both H_U and H_{SO} commute with $\mathbf{J}$. It follows that

$$[H_U + H_{\text{SO}}, J^2] = 0 \quad \text{and} \quad [H_U + H_{\text{SO}}, J_z] = 0 \quad (4.48)$$

Consequently, we can label an eigen-state of the full Hamiltonian ($H_U + H_{\text{SO}}$) with two quantum numbers: J, M_J .

In the LS -coupling scheme, we solve the the additional H_{SO} perturbatively within the H_U degenerate multiplet terms. Within each multiplet term, the eigen-states are:

$$|^{2S+1}L_{JM_J}\rangle \quad (4.49)$$

In general, within a multiplet term,

$$\boxed{|^{2S+1}L_{JM_J}\rangle = \sum_{M_S+M_L=M_J} C_{M_S M_L} |^{2S+1}L_{M_S M_L}\rangle}$$

which is a Clebsch-Gordan transformation (see Section 4.6). As a result, a multiplet term ^{2S+1}L is further divided into terms

$$^{2S+1}L_J \quad \text{where, } J = |L-S|, |L-S|+1, \dots, L+S \quad (4.50)$$

For each J ,

$$M_J = -J, -J+1, \dots, -1, 0, 1, \dots, J-1, J$$

For instance, consider the 3P term of the p^2 system. The $|^3P_{M_S M_L}\rangle$ states are:

$$\begin{array}{ccc} |^3P_{1\ 1}\rangle & |^3P_{1\ 0}\rangle & |^3P_{1\ -1}\rangle \\ |^3P_{0\ 1}\rangle & |^3P_{0\ 0}\rangle & |^3P_{0\ -1}\rangle \\ |^3P_{-1\ 1}\rangle & |^3P_{-1\ 0}\rangle & |^3P_{-1\ -1}\rangle \end{array}$$

Performing a Clebsch-Gordan transformation, we obtain the $|^3P_{JM_J}\rangle$ states:

$$\begin{array}{ccccc} |^3P_{2\ 2}\rangle & |^3P_{2\ 1}\rangle & |^3P_{2\ 0}\rangle & |^3P_{2\ -1}\rangle & |^3P_{2\ -2}\rangle \\ & |^3P_{1\ 1}\rangle & |^3P_{1\ 0}\rangle & |^3P_{1\ -1}\rangle & \\ & & |^3P_{0\ 0}\rangle & & \end{array}$$

In this manner, the 3P group splits into 3 sub-groups with distinct J :

$$^3P \text{ (9-fold degenerate)} = \begin{cases} ^3P_2 \text{ (5-fold degenerate)} \\ ^3P_1 \text{ (3-fold degenerate)} \\ ^3P_0 \text{ (1-fold degenerate)} \end{cases}$$

which is the fine-structure spin-orbit splitting within the multiplet term.

Summary:

1. We can (pictorially) understand the LS -coupling scheme as follows:
 - In the many-electron LS -basis, $\mathbf{H}_U$ is block-diagonal; $\mathbf{H}_{SO}$ is non-diagonal;
 - In the $|^{2S+1}L_{M_S M_L}\rangle$ basis, $\mathbf{H}_U$ is diagonal (up to seniority); $\mathbf{H}_{SO}$ is non-diagonal;
 - In the $|^{2S+1}L_{JM_J}\rangle$ basis, $\mathbf{H}_U$ is diagonal (up to seniority); $\mathbf{H}_{SO}$ is diagonal within each multiplet term, but it has off-diagonal elements between states from different multiplet terms. Those off-diagonal elements are presumably small if the spin-orbit interaction is indeed a perturbation.
2. $|^{2S+1}L_{M_S M_L}\rangle$ can be obtained by the ladder operator technique ($L_{\pm}$ and $S_{\pm}$). They are collected into degenerate groups called the multiplet terms:

$$^{2S+1}L \quad (2S+1)(2L+1)\text{-fold degenerate}$$

3. $|^{2S+1}L_{JM_J}\rangle$ can be obtained by the Clebsch-Gordan transformation within each

multiplet term. They form the fine-structure multiplet terms:

$$^{2S+1}L_J \quad (2J+1)\text{-fold degenerate}$$

4.5. The *jj*-coupling scheme

In the *jj*-coupling scheme, we first diagonalize H_{SO} exactly, and then solve H_U as a perturbation.

4.5.1. Diagonalizing H_{SO} in the *jj*-basis

Diagonalizing H_{SO} in the *jj*-basis is simple, because H_{SO} is *diagonal* in the *jj*-basis.

From (4.44) and (4.45), we see that H_{SO} commutes with the individual $\mathbf{j}_i$:

$$[H_{\text{SO}}, \ell_i] = -[H_{\text{SO}}, \mathbf{s}_i] \Rightarrow [H_{\text{SO}}, \mathbf{j}_i] = 0 \quad (4.51)$$

This allows an eigen-state of H_{SO} to be labelled by a set of individual quantum numbers:

$$\{j_i, \mu_i\} \quad \text{for } i \in \text{occ}$$

Thus, for each eigen-state of H_{SO} , we denote,

$$\left| (j_1, j_2 \cdots j_{N_e})_{(\mu_1, \mu_2 \cdots \mu_{N_e})} \right\rangle \quad (4.52)$$

which maps to the *jj*-basis states exactly.

For a given shell (n, l) , the spin-orbit matrix elements in the *jj*-basis (4.19) are diagonal:

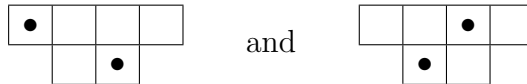
$$X_{\alpha\beta} = \frac{1}{2} \Xi_{nl} \left[j_\alpha(j_\alpha + 1) - l(l + 1) - \frac{3}{4} \right] \delta_{\alpha\beta} \quad (4.53)$$

where $\Xi_{nl} = \langle R_{nl} | \xi(r) | R_{nl} \rangle$.

For a system with N_e electrons, the eigen-energy of a *jj*-basis state reads,

$$E_{\text{SO}} = \frac{1}{2} \Xi_{nl} \sum_{i=1}^{N_e} \left[j_i(j_i + 1) - l(l + 1) - \frac{3}{4} \right] \quad (4.54)$$

The eigen-energies are only determined by the j quantum numbers (not μ) of the occupied orbitals. For instance, both (the orbital quantum numbers are given in Fig. 4.1b)



have the same energy

$$E_{\text{SO}} = \frac{1}{2} \Xi_{nl} \left[\left(\frac{3}{2} \cdot \frac{5}{2} - 2 - \frac{3}{4} \right) + \left(\frac{1}{2} \cdot \frac{3}{2} - 2 - \frac{3}{4} \right) \right] = -\frac{1}{2} \Xi_{nl} \quad (4.55)$$

Therefore, a jj -tuple

$$(j_1, j_2 \cdots j_{N_e})$$

defines a degenerate *spin-orbit term* [23]. What is its degeneracy? It is a combinatorics problem: Let N_+ be the number of j_i 's that are equal to $l + \frac{1}{2}$, and N_- be the number of j_i 's that are equal to $l - \frac{1}{2}$ (if $l = 0$, then $N_- = 0$). We have,

$$\text{degeneracy} = \binom{2l+2}{N_+} \binom{2l}{N_-} \quad (4.56)$$

Consider the jj -basis states of the p^2 system. The basis states can be grouped into degenerate terms (Fig. 4.3).

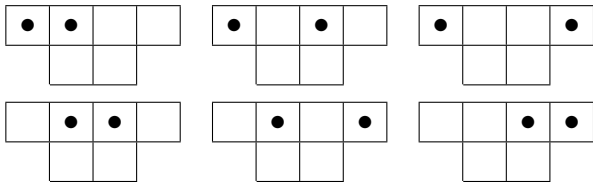
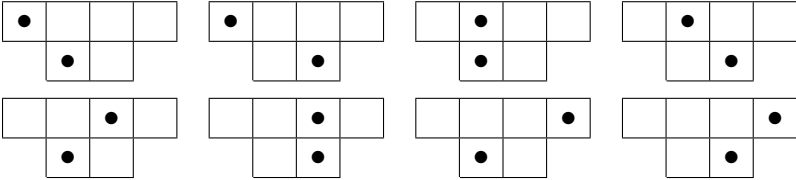
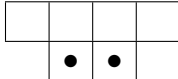
jj -tuple	Basis states (also eigen-states of H_{SO})	Term energies
$\left(\frac{3}{2}, \frac{3}{2}\right)$		Ξ_{nl}
$\left(\frac{3}{2}, \frac{1}{2}\right)$		$-\frac{1}{2}\Xi_{nl}$
$\left(\frac{1}{2}, \frac{1}{2}\right)$		$-2\Xi_{nl}$

Figure 4.3.: The many-electron basis states of a p^2 shell in the jj -coupling scheme. The basis states are grouped into degenerate terms according to the j 's of the occupied orbitals. The orbital quantum numbers are given in Fig. 4.1b.

4.5.2. Diagonalizing H_U in the jj -basis

Now, we introduce the Coulomb Hamiltonian H_U .

Because $[H_U, \mathbf{j}_i] \neq 0$ (since $[H_U, \ell_i] \neq 0$ and $[H_U, \mathbf{s}_i] = 0$), the jj -basis states are in general no longer eigen-states. But, since both H_U and H_{SO} commute with $\mathbf{J}$, an eigen-state of the full Hamiltonian ($H_U + H_{SO}$) can be labelled by two quantum numbers: J, M_J .

In the jj -coupling scheme, we solve the the additional H_U perturbatively within the H_{SO} degenerate spin-orbit terms. Within each spin-orbit term, the eigen-states are

$$|(j_1, j_2 \cdots j_{N_e})_{JM_J}\rangle \quad (4.57)$$

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

Thus, a spin-orbit term is divided into fine-structured spin-orbit terms

$$(j_1, j_2 \cdots j_{N_e})_J \quad (4.58)$$

within each term,

$$M_J = -J, -J+1, \dots, -1, 0, 1, \dots, J-1, J$$

In general, an eigen-state $|(j_1, j_2 \cdots j_{N_e})_{JM_J}\rangle$ is a linear combination of the jj -basis states with the same $(j_1, j_2 \cdots j_{N_e})$ and $\sum_i \mu_i = M_J$. Previously we used an M_S - M_L table to collect the multiplet terms. Similarly, here we use a $(j_1, j_2 \cdots j_{N_e})$ - M_J table to collect the fine-structure spin-orbit terms. Consider the p^2 system:

		M_J				
		2	1	0	-1	-2
(j_1, j_2)	$\left(\frac{3}{2}, \frac{3}{2}\right)$	1	1	2	1	1
	$\left(\frac{3}{2}, \frac{1}{2}\right)$	1	2	2	2	1
	$\left(\frac{1}{2}, \frac{1}{2}\right)$	0	0	1	0	0

Each row stands for one spin-orbit term subspace. Within each subspace, we can construct eigen-states of H_U using the ladder operator technique ($J_{\pm}$). For the p^2 system, the groups are:

$$\begin{pmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & \left(\frac{3}{2}, \frac{3}{2}\right)_2 + \left(\frac{3}{2}, \frac{3}{2}\right)_0 \\ + \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & \left(\frac{3}{2}, \frac{1}{2}\right)_2 + \left(\frac{3}{2}, \frac{1}{2}\right)_1 \\ + \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} & \left(\frac{1}{2}, \frac{1}{2}\right)_0 \end{cases}$$

Using the ladder operator technique and the orthogonalization relations, we can construct the 15 eigen-states:

$\left \left(\frac{3}{2}, \frac{3}{2} \right)_2 \ 2 \right\rangle =$	$c_{\frac{3}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger}$	$ 0\rangle$	
$\left \left(\frac{3}{2}, \frac{3}{2} \right)_2 \ 1 \right\rangle =$	$c_{\frac{3}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger}$	$ 0\rangle$	
$\left \left(\frac{3}{2}, \frac{3}{2} \right)_2 \ 0 \right\rangle =$	$\left(\frac{\sqrt{2}}{2} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} + \frac{\sqrt{2}}{2} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} \right)$	$ 0\rangle$	$\left \left(\frac{3}{2}, \frac{3}{2} \right)_0 \ 0 \right\rangle = \left(\frac{\sqrt{2}}{2} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} - \frac{\sqrt{2}}{2} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} \right) 0\rangle$
$\left \left(\frac{3}{2}, \frac{3}{2} \right)_2 \ -1 \right\rangle =$	$c_{\frac{3}{2} \frac{3}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger}$	$ 0\rangle$	
$\left \left(\frac{3}{2}, \frac{3}{2} \right)_2 \ -2 \right\rangle =$	$c_{\frac{3}{2} \frac{3}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger}$	$ 0\rangle$	
$\left \left(\frac{3}{2}, \frac{1}{2} \right)_2 \ 2 \right\rangle =$	$c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger}$	$ 0\rangle$	
$\left \left(\frac{3}{2}, \frac{1}{2} \right)_2 \ 1 \right\rangle =$	$\left(\frac{1}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} + \frac{\sqrt{3}}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} \right)$	$ 0\rangle$	$\left \left(\frac{3}{2}, \frac{1}{2} \right)_1 \ 1 \right\rangle = \left(\frac{\sqrt{3}}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} - \frac{1}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} \right) 0\rangle$
$\left \left(\frac{3}{2}, \frac{1}{2} \right)_2 \ 0 \right\rangle =$	$\left(\frac{\sqrt{2}}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} + \frac{\sqrt{2}}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} \right)$	$ 0\rangle$	$\left \left(\frac{3}{2}, \frac{1}{2} \right)_1 \ 0 \right\rangle = \left(\frac{\sqrt{2}}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} - \frac{\sqrt{2}}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} \right) 0\rangle$
$\left \left(\frac{3}{2}, \frac{1}{2} \right)_2 \ -1 \right\rangle =$	$\left(\frac{\sqrt{3}}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} + \frac{1}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} \right)$	$ 0\rangle$	$\left \left(\frac{3}{2}, \frac{1}{2} \right)_1 \ -1 \right\rangle = \left(\frac{1}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{1}{2}}^{\dagger} - \frac{\sqrt{3}}{2} c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger} \right) 0\rangle$
$\left \left(\frac{3}{2}, \frac{1}{2} \right)_2 \ -2 \right\rangle =$	$c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{3}{2} \frac{3}{2}}^{\dagger}$	$ 0\rangle$	
$\left \left(\frac{1}{2}, \frac{1}{2} \right)_0 \ 0 \right\rangle =$	$c_{\frac{1}{2} \frac{1}{2}}^{\dagger} c_{\frac{1}{2} \frac{1}{2}}^{\dagger}$	$ 0\rangle$	

where, for a better alignment, I used a notation $c_{j\bar{\mu}}^{\dagger} \leftarrow c_{j,-\mu}^{\dagger}$ (as if lifting the minus sign to the top, sorry for the confusion that it looks like the complex conjugate symbol).

For general open-shell systems, we set up the many-electron basis using Algorithm 4.3. The **BasisJJ** class (similar to the **BasisLS** class) inherits from the general **Basis** class (Algorithm 1.1). Algorithm 4.4 produces the spin-orbit terms for open-shell systems with given shell number l and the number of electrons N_e .

Algorithm 4.3: The **BasisJJ** class (inherit from **Basis** (Algorithm 1.1))

```

1 class BasisJJ(Basis):
2     def __init__(self, l, Ne):
3         Basis.__init__(self, 4*l+2, Ne)
4         self.l = l # Shell l
5
6         # Quantum numbers of individual orbitals
7         self.j = [l+0.5-a//(2*l+2) for a in range(self.Norb)] # Orbital j
8         self.mj = [self.j[a]-a%(2*l+2) for a in range(self.Norb)] # Orbital mj
9
10        # Quantum numbers of many-electron basis
11        self.jj = [self.getjj(iconf) for iconf in self.conf] # Basis jj-tuple
12        self.MJ = [self.getMJ(iconf) for iconf in self.conf] # Basis total MJ
13
14        # Collect jj-tuple of iconf
15        def getjj(self, iconf):
16            return tuple(self.j[a] for a in range(self.Norb) if isBit(iconf, a))
17
18        # Calculate total MJ of iconf
19        def getMJ(self, iconf):
20            return sum(self.mj[a] for a in range(self.Norb) if isBit(iconf, a))

```

4. *LS-coupling and jj-coupling schemes for atomic open-shell problems*

```

21
22 # Apply J- to a vector (in the many-electron basis)
23 def Jm(self, v):
24     w = np.zeros(self.dim)
25     # Apply J- to non-zero basis states
26     for i in np.nonzero(v)[0]:
27         iconf = self.conf[i]
28         # Orbitals with mj > -j
29         for a in list(range(self.Norb//2))+list(range(self.Norb//2+1, self.Norb-1)):
30             # Apply j-
31             b = a + 1
32             if isBit(iconf, a) and not isBit(iconf, b):
33                 jconf = setBit(clearBit(iconf, a), b)
34                 j = self.index[jconf]
35                 alpha = math.sqrt((self.j[a]+self.mj[a])*(self.j[a]-self.mj[a]+1))
36                 w[j] += alpha*v[i] # fsign = 1
37     return w/np.linalg.norm(w)

```

Algorithm 4.4: Construct spin-orbit terms for an open-shell system

```

1 from basis import BasisJJ
2 import numpy as np
3
4 def spinorbit_terms(l, Ne):
5     # Set up basis
6     B = BasisJJ(l, Ne)
7
8     # Set up jj-MJ table (jj are tuples, MJ can be half-integer)
9     jjList = sorted(set(B.jj), reverse=True)
10    MJList = sorted(set(B.MJ), reverse=True)
11    jjMJ = list(zip(B.jj, B.MJ)) # Basis (jj, MJ) tuples
12    table = {jj: {MJ: jjMJ.count((jj,MJ)) for MJ in MJList} for jj in jjList}
13
14    # Scan the table and collect spin-orbit terms
15    terms = []
16    for jj in jjList:
17        for J in MJList[:len(MJList)//2+1]:
18            N = table[jj][J]
19            if N>0:
20                for MJ in np.arange(-J, J+1):
21                    table[jj][MJ] -= N
22                    terms.append((jj, J, N))
23
24    # Construct spin-orbit eigen-vectors
25    vecs = {}
26    for (jj, J, N) in terms:
27        for sn in range(N):
28            vecs[(jj,J,sn)] = {MJ: np.zeros(B.dim) for MJ in np.arange(-J, J+1)}
29
30    # Prepare (N-number-of) leading vectors (leading means MJ=J)
31    if N==1 and jjMJ.count((jj,J))==1:
32        # Unique basis state
33        vecs[(jj,J,0)][J][jjMJ.index((jj,J))] = 1
34    else:
35        # Previous vecs with MJ=J
36        P = [pv[J] for ((pjj,pJ,psn),pv) in vecs.items() if pjj==jj and pJ>J]
37        # Solve an SVD problem to complete the orthonormal basis
38        nz = [jjMJi==(jj,J) for jjMJi in jjMJ]
39        A = [p[nz] for p in P]
40        vh = np.linalg.svd(A)[2]
41        for sn in range(N):
42            vecs[(jj,J,sn)][J][nz] = vh[len(P)+sn]
43
44    # Ladder operator technique
45    for sn in range(N):

```

```

46     for MJ in np.arange(J, -J, -1):
47         vecs[(jj,J,sn)][MJ-1] = B.Jm(vecs[(jj,J,sn)][MJ]) # Apply J-
48     return (terms, vecs, B)

```

The function `spinorbit_terms` works in the way same as the function `multiplet_terms` (Algorithm 4.2). In the return, `terms` is a list of fine-structured spin-orbit term names. Each term in `terms` is a tuple $((j_1, j_2 \cdots j_{N_e}), J, N)$, which represents: $N \times (j_1, j_2 \cdots j_{N_e})_J$, where N is the number of seniorities. `vecs` is the collection of the (eigen-)vectors (constructed by the ladder operators) in the basis B .

Some examples: (only print out the `terms`)

```
>>> spinorbit_terms(1, 2)[0]
```

$$\left(\frac{3}{2}, \frac{3}{2}\right)_2, \left(\frac{3}{2}, \frac{3}{2}\right)_0, \left(\frac{3}{2}, \frac{1}{2}\right)_2, \left(\frac{3}{2}, \frac{1}{2}\right)_1, \left(\frac{1}{2}, \frac{1}{2}\right)_0$$

That is the p^2 system – agrees with what we have discussed.

```
>>> spinorbit_terms(2, 3)[0]
```

$$\begin{aligned}
& \left(\frac{5}{2}, \frac{5}{2}, \frac{5}{2}\right)_{\frac{9}{2}}, \quad \left(\frac{5}{2}, \frac{5}{2}, \frac{5}{2}\right)_{\frac{5}{2}}, \quad \left(\frac{5}{2}, \frac{5}{2}, \frac{5}{2}\right)_{\frac{3}{2}}, \quad \left(\frac{5}{2}, \frac{5}{2}, \frac{3}{2}\right)_{\frac{11}{2}}, \quad \left(\frac{5}{2}, \frac{5}{2}, \frac{3}{2}\right)_{\frac{9}{2}}, \\
& 2 \times \left(\frac{5}{2}, \frac{5}{2}, \frac{3}{2}\right)_{\frac{7}{2}}, \quad 2 \times \left(\frac{5}{2}, \frac{5}{2}, \frac{3}{2}\right)_{\frac{5}{2}}, \quad 2 \times \left(\frac{5}{2}, \frac{5}{2}, \frac{3}{2}\right)_{\frac{3}{2}}, \quad \left(\frac{5}{2}, \frac{5}{2}, \frac{3}{2}\right)_{\frac{1}{2}}, \quad \left(\frac{5}{2}, \frac{3}{2}, \frac{3}{2}\right)_{\frac{9}{2}}, \\
& \left(\frac{5}{2}, \frac{3}{2}, \frac{3}{2}\right)_{\frac{7}{2}}, \quad 2 \times \left(\frac{5}{2}, \frac{3}{2}, \frac{3}{2}\right)_{\frac{5}{2}}, \quad \left(\frac{5}{2}, \frac{3}{2}, \frac{3}{2}\right)_{\frac{3}{2}}, \quad \left(\frac{5}{2}, \frac{3}{2}, \frac{3}{2}\right)_{\frac{1}{2}}, \quad \left(\frac{3}{2}, \frac{3}{2}, \frac{3}{2}\right)_{\frac{3}{2}}
\end{aligned}$$

That is the d^3 system. Similar to the multiplet terms, because of H_U , there are also “seniority cases” for the fine-structured spin-orbit terms that different terms can have the same angular momenta.

Summary:

1. We can (pictorially) understand the jj -coupling scheme as follows:
 - In the many-electron jj -basis, $\mathbf{H}_{\text{SO}}$ is diagonal; $\mathbf{H}_U$ is non-diagonal;
 - In the $|(j_1, j_2 \cdots j_{N_e})_{JM_J}\rangle$ basis, $\mathbf{H}_{\text{SO}}$ is diagonal; $\mathbf{H}_U$ is diagonal (up to seniority) within each spin-orbit term, but it has off-diagonal elements between states from different spin-orbit terms. Those off-diagonal elements are presumably small if the Coulomb interaction is indeed a perturbation (rare case).
2. The jj -basis states can be collected into degenerate groups called the spin-orbit terms:

$$(j_1, j_2 \cdots j_{N_e}) \quad \binom{2l+2}{N_+} \binom{2l}{N_-} \text{-fold degenerate}$$

3. $|(j_1, j_2 \cdots j_{N_e})_{JM_J}\rangle$ can be obtained by the ladder operator technique ($J_{\pm}$) within each spin-orbit term. They form the fine-structure spin-orbit terms:

$$(j_1, j_2 \cdots j_{N_e})_J \quad (2J+1)\text{-fold degenerate}$$

4.6. Transformation between *LS*-basis and *jj*-basis states

In this section, we will discuss how to transform a many-body state given in the *jj*-basis in terms of the *LS*-basis states, and vice versa. The transformation is useful and sometimes necessary when computing the matrix elements. In particular, the evaluation of the Coulomb matrix element requires explicitly integrating over the real space coordinates $\mathbf{r}_1$ and $\mathbf{r}_2$, where the wave functions R_{nl} and Y_{lm} are used. The wave functions are the orbitals in the *LS*-basis. Thus, the computation of the Coulomb matrix elements has to be performed in the *LS*-basis.

In the previous sections, we used the notations $c_\alpha^\dagger$ and c_α for the creation and annihilation operators in both the *LS*- and the *jj*-coupling schemes. In this section, to better distinguish the two bases, for a given shell, I use

$$\begin{aligned} c_{m\sigma}^\dagger, c_{m\sigma} & \text{ for } LS\text{-basis orbitals} \\ d_{j\mu}^\dagger, d_{j\mu} & \text{ for } jj\text{-basis orbitals} \end{aligned}$$

The one-electron orbitals in the *LS*- and the *jj*-basis are related by the Clebsch-Gordan transformation. For a review purpose, we do a small exercise.

Exercise:

Suppose we have an electron in a *p*-shell. Express the one-electron state $d_{j\mu}^\dagger |0\rangle$ in terms of $c_{m\sigma}^\dagger |0\rangle$.

Solution:

In a *p*-shell ($l = 1$ and $s = \frac{1}{2}$), the possible *j*'s are $\frac{3}{2}$ and $\frac{1}{2}$.

We start from the leading state $d_{j\mu}^\dagger |0\rangle = d_{\frac{3}{2}\frac{3}{2}}^\dagger |0\rangle$. Among the $c_{m\sigma}^\dagger |0\rangle$ states, the only state that fulfills $m + \sigma = \mu = \frac{3}{2}$ is $c_{1\uparrow}^\dagger |0\rangle$. Hence,

$$d_{\frac{3}{2}\frac{3}{2}}^\dagger |0\rangle = c_{1\uparrow}^\dagger |0\rangle \quad (4.59)$$

The subsequent states can be obtained by applying the lowering ladder operators:

$$J_- d_{\frac{3}{2}\frac{3}{2}}^\dagger |0\rangle = (L_- + S_-) c_{1\uparrow}^\dagger |0\rangle \Rightarrow d_{\frac{3}{2}\frac{1}{2}}^\dagger |0\rangle = \sqrt{\frac{2}{3}} c_{0\uparrow}^\dagger |0\rangle + \sqrt{\frac{1}{3}} c_{1\downarrow}^\dagger |0\rangle \quad (4.60)$$

Keep applying J_- , we obtain,

$$d_{\frac{3}{2},-\frac{1}{2}}^\dagger |0\rangle = \sqrt{\frac{1}{3}} c_{-1\uparrow}^\dagger |0\rangle + \sqrt{\frac{2}{3}} c_{0\downarrow}^\dagger |0\rangle \quad (4.61)$$

$$d_{\frac{3}{2},-\frac{3}{2}}^\dagger |0\rangle = c_{-1\downarrow}^\dagger |0\rangle \quad (4.62)$$

4.6. Transformation between LS -basis and jj -basis states

The next leading state is $d_{j\mu}^\dagger |0\rangle = d_{\frac{1}{2}\frac{1}{2}}^\dagger |0\rangle$. It is related to $d_{\frac{3}{2}\frac{1}{2}}^\dagger |0\rangle$ by the same $\mu = \frac{1}{2}$. Using the orthogonalization relation, we have, (here enters a phase convention, since both $d_{\frac{1}{2}\frac{1}{2}}^\dagger |0\rangle$ and $-d_{\frac{1}{2}\frac{1}{2}}^\dagger |0\rangle$ are orthogonal to $d_{\frac{3}{2}\frac{1}{2}}^\dagger |0\rangle$.)

$$d_{\frac{1}{2}\frac{1}{2}}^\dagger |0\rangle = \sqrt{\frac{1}{3}} c_{0\uparrow}^\dagger |0\rangle - \sqrt{\frac{2}{3}} c_{1\downarrow}^\dagger |0\rangle \quad (4.63)$$

Applying J_- , we get,

$$d_{\frac{1}{2},-\frac{1}{2}}^\dagger |0\rangle = \sqrt{\frac{2}{3}} c_{-1\uparrow}^\dagger |0\rangle - \sqrt{\frac{1}{3}} c_{0\downarrow}^\dagger |0\rangle \quad (4.64)$$

Done! We have expressed all $d_{j\mu}^\dagger |0\rangle$ as linear combinations of $c_{m\sigma}^\dagger |0\rangle$:

$$d_{j\mu}^\dagger |0\rangle = \sum_{m,\sigma} c_{m\sigma}^\dagger |0\rangle \langle 0| c_{m\sigma} d_{j\mu}^\dagger |0\rangle \quad (4.65)$$

where $\langle 0| c_{m\sigma} d_{j\mu}^\dagger |0\rangle$ are the Clebsch-Gordan coefficients. Notice that they are evaluated not from explicit integrals but using the ladder operator technique (an example code is given in Algorithm 4.5). Equivalently, we can relate the electron creators of the two bases as

$$d_{j\mu}^\dagger = \sum_{m,\sigma} \langle 0| c_{m\sigma} d_{j\mu}^\dagger |0\rangle c_{m\sigma}^\dagger \quad (4.66)$$

Now, we can transform the many-body states. For example, consider the jj -basis state

$$| \begin{array}{|c|c|c|c|} \hline \bullet & & & \\ \hline & & & \\ \hline & & \bullet & \\ \hline \end{array} \rangle = d_{\frac{1}{2},-\frac{1}{2}}^\dagger d_{\frac{3}{2}\frac{3}{2}}^\dagger |0\rangle$$

According to (4.59) and (4.64), we have

$$d_{\frac{1}{2},-\frac{1}{2}}^\dagger d_{\frac{3}{2}\frac{3}{2}}^\dagger |0\rangle = \left(\sqrt{\frac{2}{3}} c_{-1\uparrow}^\dagger - \sqrt{\frac{1}{3}} c_{0\downarrow}^\dagger \right) \left(c_{1\uparrow}^\dagger \right) |0\rangle = \left(\sqrt{\frac{2}{3}} c_{-1\uparrow}^\dagger c_{1\uparrow}^\dagger - \sqrt{\frac{1}{3}} c_{0\downarrow}^\dagger c_{1\uparrow}^\dagger \right) |0\rangle \quad (4.67)$$

In our configuration-diagram language, Eqn. (4.67) says,

$$| \begin{array}{|c|c|c|c|} \hline \bullet & & & \\ \hline & & & \\ \hline & & \bullet & \\ \hline \end{array} \rangle = \sqrt{\frac{2}{3}} | \begin{array}{|c|c|c|c|} \hline \bullet & & \bullet & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \rangle - \sqrt{\frac{1}{3}} | \begin{array}{|c|c|c|c|} \hline \bullet & & & \\ \hline & & & \\ \hline & & \bullet & \\ \hline \end{array} \rangle \quad (4.68)$$

This is how we express a jj -basis state in terms of LS -basis states. Suppose (for a given system) we have transformed all the jj -basis states into the LS -basis states. We collect all the vectors (as columns) in a matrix:

$$\mathbf{U} = [\mathbf{v}_1 | \mathbf{v}_2 | \cdots | \mathbf{v}_{\dim_H}] \quad (4.69)$$

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

where $\mathbf{v}_i$ is the vector representation of the i -th jj -basis state in the LS -basis. Now, with this $\mathbf{U}$ matrix (unitary), we can transform an arbitrary state between LS - and jj -basis representations straightforwardly:

$$\mathbf{v}_{LS} = \mathbf{U} \mathbf{v}_{jj} \quad (4.70)$$

$$\mathbf{v}_{jj} = \mathbf{U}^\dagger \mathbf{v}_{LS} \quad (4.71)$$

where $\mathbf{v}_{LS}$ and $\mathbf{v}_{jj}$ are the two representations of the *same* state in the LS -basis and the jj -basis, respectively. Certainly, this unitary transformation also enables us to transform matrix representations between the two bases:

$$\mathbf{H}_{LS} = \mathbf{U} \mathbf{H}_{jj} \mathbf{U}^\dagger \quad (4.72)$$

$$\mathbf{H}_{jj} = \mathbf{U}^\dagger \mathbf{H}_{LS} \mathbf{U} \quad (4.73)$$

Algorithm 4.5: Compute Clebsch-Gordan coefficients using the ladder operator technique (cg.py).

```

1  from math import sqrt
2  import numpy as np
3
4  # Clebsch-Gordan table (j1, j2 must be non-negative int or half-int)
5  def CG(j1, j2):
6      Jmax = j1+j2; Jmin = abs(j1-j2)
7
8      # (J,M)-basis (row index)
9      JM = [(J,M) for J in np.arange(Jmax, Jmin-1, -1) for M in np.arange(J, -J-1, -1)]
10     # (m1,m2)-basis (col index)
11     mm = [(m1,m2) for m1 in np.arange(j1, -j1-1, -1) for m2 in np.arange(j2, -j2-1, -1)]
12
13     # CG table
14     T = {JMi: {mmj: 0 for mmj in mm} for JMi in JM}; T[(Jmax, Jmax)][(j1, j2)] = 1.0
15     for J in np.arange(Jmax, Jmin-1, -1):
16         # Leading |J J>
17         if J < Jmax:
18             # Previous |Jmax J> ... |J+1 J>
19             P = [T[(Jp, M)] for (Jp, M) in JM if Jp > J and M == J]
20             # Make |J J> orthogonal to |Jmax J> ... |J+1 J>
21             mmJ = [(m1, m2) for (m1, m2) in mm if m1+m2 == J]
22             A = [[p[mmj] for mmj in mmJ] for p in P]
23             v = np.linalg.svd(A)[2][-1]
24             v *= np.sign(v[mmJ.index((j1, J-j1))]) # Phase convention
25             for (j, mmj) in enumerate(mmJ): T[(J, J)][mmj] = v[j]
26
27     # Apply J- = j1- + j2- to get subsequent |J J-1> ... |J -J>
28     for M in np.arange(J, -J, -1):
29         mm1 = ((m1, m2) for (m1, m2) in mm if m1+m2 == M and m1 > -j1)
30         mm2 = ((m1, m2) for (m1, m2) in mm if m1+m2 == M and m2 > -j2)
31         for (m1, m2) in mm1:
32             T[(J, M-1)][(m1-1, m2)] += T[(J, M)][(m1, m2)]*sqrt((j1+m1)*(j1-m1+1)/(J+M)/(J-M+1))
33         for (m1, m2) in mm2:
34             T[(J, M-1)][(m1, m2-1)] += T[(J, M)][(m1, m2)]*sqrt((j2+m2)*(j2-m2+1)/(J+M)/(J-M+1))
35     return T

```

4.7. The intermediate-coupling scheme

The intermediate-coupling scheme is computationally more expensive, but conceptually, it is the most straightforward scheme. In the LS -coupling scheme, H_U is treated exactly and H_{SO} is a perturbation within the multiplet terms; In the jj -coupling scheme, H_{SO} is treated exactly and H_U is a perturbation within spin-orbit terms. Both LS - and jj -coupling schemes involve perturbative approximations such that analytical solutions are possible. In the intermediate-coupling scheme, the Coulomb H_U and the spin-orbit H_{SO} are treated on an equal footing: we diagonalize the full Hamiltonian ($H_U + H_{SO}$) exactly.

As we have discussed, the full Hamiltonian ($H_U + H_{SO}$) does not commute with $\mathbf{L}$, $\mathbf{S}$, or the individual $\mathbf{j}_i$. It only commutes with the total $\mathbf{J}$. Consequently, the eigen-states of ($H_U + H_{SO}$) are labelled as $|J, M_J\rangle$. The hope is that we count the M_J of the basis states and construct eigen-states by the ladder operator $J_{\pm}$.

If we choose the LS -basis, we calculate $M_J = \sum_i (m_i + \sigma_i)$; If we choose the jj -basis, we calculate $M_J = \sum_i \mu_i$. Independent of the choice of basis, for a given open-shell system, we should obtain the same M_J -table counted from the basis states. For instance, for the simple p^2 system, one gets

M_J	2	1	0	-1	-2
#-of-states	2	3	5	3	2

Now, we can collect the groups of eigen-states that are connected by $J_{\pm}$:

$$(2 \ 3 \ 5 \ 3 \ 2) = \left\{ \begin{array}{ll} \left(\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \end{array} \right) & \rightarrow 2 \times (J = 2) \\ + \\ \left(\begin{array}{ccccc} 0 & 1 & 1 & 1 & 0 \end{array} \right) & \rightarrow (J = 1) \\ + \\ \left(\begin{array}{ccccc} 0 & 0 & 2 & 0 & 0 \end{array} \right) & \rightarrow 2 \times (J = 0) \end{array} \right.$$

We are in a situation that the “seniority trouble” ($N = 2$) occurs from the beginning. We cannot start from a unique leading state to apply the ladder operators (one can obtain the two leading states numerically by diagonalizing a 2-by-2 matrix and continue applying the ladder operators). Embarrassingly, not a single eigen-state can be solved analytically, even for the simple p^2 system. That is the fundamental issue with the intermediate-coupling scheme – the eigen-states are *not* analytical. However, in terms of numerical solutions, the intermediate-coupling scheme produces the best results since both H_U and H_{SO} are treated exactly. As a remark, in the intermediate-coupling scheme, the eigen-energies of the Hamiltonians are independent of the choice of basis: both the LS -basis and the jj -basis span the same Hilbert space.

At this stage, the three coupling schemes are discussed. We are in a position to have a concrete example. In the final part of Chapter 3, we had an example of the DFT calculation of a lead atom ($s_2\text{Pb}$) with open-shell $6p^2$. The lead atom is an ideal candidate

4. LS -coupling and jj -coupling schemes for atomic open-shell problems

for illustrating the concepts of the three coupling schemes, for two reasons: (1) It has the simplest non-trivial open-shell p^2 system; (2) It is sufficiently heavy (with atomic number $Z = 82$), so that the spin-orbit interaction is comparable with the Coulomb interaction.

To calculate realistic matrix elements $\ddot{U}_{\alpha\beta\gamma\delta}$ and $X_{\alpha\beta}$ for the lead $6p^2$ open-shell system, we use the self-consistent radial wave function $R_{6p}(r)$ and the self-consistent mean-field potential $V_{KS}(r)$ from the DFT calculations. That is the “link” between the two chapters: The DFT results of the atoms prepare the realistic orbital basis for the many-body calculations.

The final results are summarized in Fig. 4.4. The left three columns describe the LS -coupling scheme: the degenerate $6p^2$ shell splits into multiplet terms due to the effect of the Coulomb interaction; within each term, the energy levels split further due to the spin-orbit perturbation. The right three columns describe the jj -coupling scheme: the degenerate $6p^2$ shell splits into spin-orbit terms due to the effect of the spin-orbit interaction; within each term, the energy levels split further due to the Coulomb perturbation. The column in the middle displays the numerical results from the intermediate-coupling scheme, which serves as a reference judging the quality of the LS - and jj -perturbative approximations. The multiplet and spin-orbit energy splittings are of the order of 10 milli-Hartree. Those energy gaps are experimentally measurable [24], and we will compare with in the next chapter. Recall the energy gaps among the *shells* (Table 3.3), where the energy gaps are of the order of 1 Hartree, which is two orders of magnitude greater than the multiplet and spin-orbit splittings. That is the fundamental reason that in most of the systems, to a good approximation, we can focus on one specific open-shell without taking into account of the mixture among many shells.

As a remark, in the lead $6p^2$ open-shell, the strength of the Coulomb and spin-orbit interactions are of the same order of magnitude. As a result, neither the LS - nor the jj -coupling schemes are good approximations for the lead $6p^2$ open-shell, because it is unfair to treat either of the two interactions as a perturbation. The resulting energy levels computed from the two schemes have similar qualities and they differ from the intermediate-coupling scheme results by 30% $\sim$ 40% in terms of the spectra *variance*. The LS -coupling scheme works significantly better for lighter elements, like ${}_6\text{C}$, ${}_{14}\text{Si}$, ${}_{32}\text{Ge}$, and ${}_{50}\text{Sn}$. The jj -coupling scheme works better for super heavy elements like ${}_{114}\text{Fl}$. Those periodic trends of the many-body effects will be discussed systematically in the next chapter.

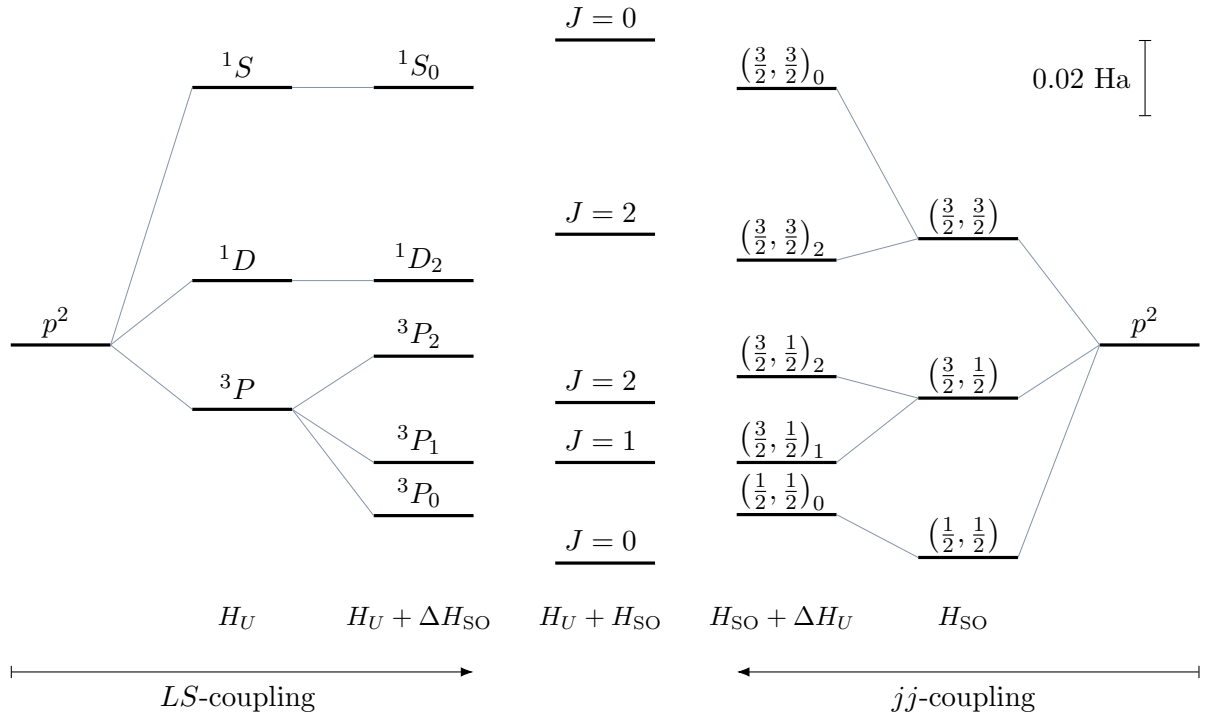


Figure 4.4.: Comparison between the LS -coupling scheme and the jj -coupling scheme for a lead (82Pb) atom $6p^2$ open-shell. Left three columns: LS -coupling scheme; Right three columns: jj -coupling scheme; Middle column: Intermediate-coupling scheme.

5. Trends of many-body effects in atomic open-shells

5.1. Trends of the matrix elements

In the previous chapter, we discussed solving the *many-body* problem for atomic open-shell systems. We formulated the problems using the *LS*- and *jj*-coupling schemes so that we could solve the many-body problems analytically in a perturbative approach. As for which coupling scheme gives a more accurate result, one must assess the relative importance between the Coulomb and spin-orbit interaction for the given system. In this chapter, we are going to study systematically the trends of the two interactions in the atomic open-shell systems over the entire periodic table.

The building blocks of the many-body Hamiltonians H_U and H_{SO} are the matrix elements $U_{\alpha\beta\gamma\delta}$ (4.12) and $X_{\alpha\beta}$ (4.13). The matrix elements are the actual medium where the specific physical information is stored. The study of the trends of the two interactions is to a certain degree the study of the trends of the matrix elements. As we saw from Eqns. (4.12), (4.14), and (4.17), the matrix elements can be written as products of radial and angular components. The angular part of the matrix elements are universal, in the sense that for shells of the same type (same l), the angular wave functions are identical. On the other hand, to obtain the radial part of the matrix elements, we use the self-consistent radial wave functions and the mean-field potentials from the DFT calculations, which are system dependent. Therefore, our focus is to study the periodic trends of the *radial* components of the matrix elements. In particular, we study the atomic open-shell **Slater-Condon parameters** $F_{nl}^{(k)}$ and the **spin-orbit parameters** Ξ_{nl} :

$$F_{nl}^{(k)} = (R_{nl}R_{nl} | \frac{r_{<}^k}{r_{>}^{k+1}} | R_{nl}R_{nl}) \quad (5.1)$$

$$\Xi_{nl} = \langle R_{nl} | \xi(r) | R_{nl} \rangle \quad (5.2)$$

where R_{nl} (or u_{nl}) are the atomic orbital radial wave functions in the open-shell; $r_{<} = \min(r_1, r_2)$ and $r_{>} = \max(r_1, r_2)$; The form of the two-electron integral in (5.1) is given in (1.18); $\xi(r)$ is the $\ell \cdot s$ proportionality term [21, 25] given in (4.3).

To study the trends, we start our discussion from understanding the general *pattern* of the electronic configurations across the periodic table.

5.2. DFT calculations and the structure of the periodic table

When performing a DFT calculation for an atomic system, the first step is to specify the electronic configuration (see Algorithm 3.7). In other words, one has to decide which orbitals are filled and which ones are empty. In fact, it is **not** a trivial question to ask, *what is the ground state configuration of a many-body system?*

Many-body systems are practically unsolvable, hence the ground state configurations are analytically unknown. Experimentally, the ground state configurations can be determined by the examination of atomic spectra [26]. Computationally, through DFT calculations, we can calculate the total energy of the system and deduce the ground state configuration by varying the occupations. Consider a calcium atom ($_{20}\text{Ca}$), suppose its ground state configuration is unknown, we perform DFT calculations with different configurations, as shown in Table 5.1.

Table 5.1.: DFT (LDA) calculations of a calcium atom ($_{20}\text{Ca}$) with different electronic configurations.

[Ar] $4s^0 3d^2$		[Ar] $4s^1 3d^1$		[Ar] $4s^2 3d^0$	
$3d^2$	-0.029299	$3d^1$	-0.032338	$3d^0$	-0.083078
$4s^0$	-0.126284	$4s^1$	-0.123205	$4s^2$	-0.141411
$3p^6$	-0.924085	$3p^6$	-0.944030	$3p^6$	-1.030573
$3s^2$	-1.592119	$3s^2$	-1.614527	$3s^2$	-1.706331
$2p^6$	-12.157289	$2p^6$	-12.182386	$2p^6$	-12.285376
$2s^2$	-14.915776	$2s^2$	-14.941971	$2s^2$	-15.046906
$1s^2$	-143.834285	$1s^2$	-143.849655	$1s^2$	-143.935181
E_{tot}	-675.569058	E_{tot}	-675.663819	E_{tot}	-675.742283

Evidently, the $4s$ orbital has a lower energy than the $3d$ (in contrast to the hydrogen-like systems), and the $4s$ orbital should be filled before the $3d$. The electronic configuration $[\text{Ar}] 4s^2$ leads to the ground state of the calcium atom ($_{20}\text{Ca}$).

In hydrogen-like systems, the orbital energies are simply determined by the principal quantum number n (but not l), which is a consequence (and also a coincidence) of solving the Schrödinger equation (3.1) with the nuclear potential having the $1/r$ form. However, solving the many-body systems in the Kohn-Sham approach (3.33) replaces the nuclear potential by the effective Kohn-Sham potential $V_{\text{KS}}(r)$, which immediately breaks the energy degeneracy among different l states for a given n . Empirically, the filling order for ground state configurations is summarized by the Aufbau principle with Madelung's rule: [27, 28]

1. Orbitals are filled in the order of increasing $n + l$;
2. In the case of equal $n + l$ values, the orbital with a lower n is filled first.

5.2. DFT calculations and the structure of the periodic table

This gives the following order for filling the orbitals:

$$1s, 2s, 2p, 3s, 3p, 4s, 3d, 4p, 5s, 4d, 5p, 6s, 4f, 5d, 6p, 7s, 5f, 6d, 7p, \dots$$

Following the ordering, given that an s -shell can be filled with 2 electrons, a p -shell can be filled with 6 electrons, a d -shell can be filled with 10 electrons, an f -shell can be filled with 14 electrons, etc., the entire periodic table arranges automatically into different blocks according to where the “last” electron resides, as shown in Fig. 5.1. Charles Janet [29] used the idea and arranged the “left-step periodic table” [30, 31], which may better represent the shell filling sequence and the block structures.

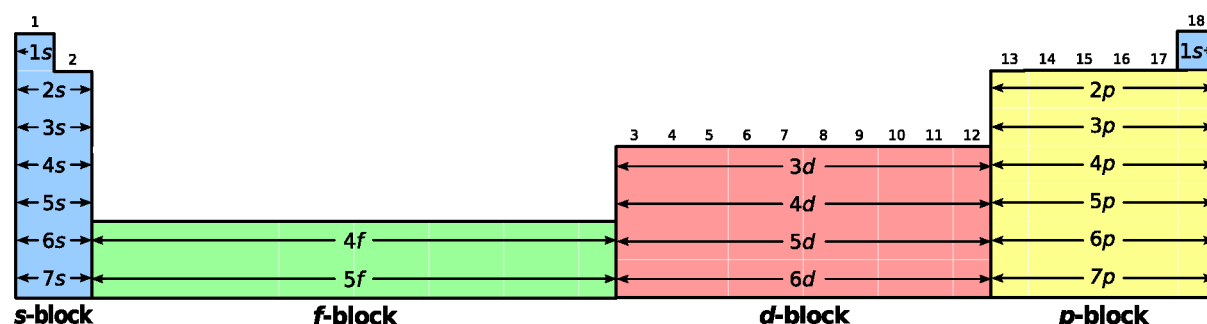


Figure 5.1.: The s, p, d, f block arrangement of the periodic table.

The block-structured periodic table shown in Fig. 5.1 is arranged *ideally* according to the Aufbau principle with Madelung’s rule. The s -block consists of groups 1–2 (IUPAC group numbering [32]) plus helium; The p -block consists of groups 13–18 excluding helium; The d -block comprises groups 3–12 (concerning the ambiguity of group 3, see [33]); The f -block is often offset below (with no group numbers), since it makes the entire periodic table as wide as shown in Fig. 5.1.

From an experimental point of view, the Aufbau principle with Madelung’s rule agrees with the ground state configurations for most of the neutral atoms, with only a handful of exceptions in transition metals, lanthanides, and actinides. For example, the ground state configuration of copper ($_{29}\text{Cu}$) is measured [26] as $[\text{Ar}] 4s^1 3d^{10}$, instead of Madelung’s $[\text{Ar}] 4s^2 3d^9$. Table 5.2 lists the 20 exceptions among the neutral atoms. Note that the configurations of elements beyond hassium ($_{108}\text{Hs}$) have not yet been experimentally verified [26] (as the elements are extremely short-lived), but they are predicted to follow Madelung’s rule without exceptions until element 120 (see Reference [34]).

The Aufbau principle with Madelung’s rule describes the general pattern of the ground state configurations of *neutral* atoms. The atoms that are *ionized* (oxidized), on the other hand, follows a different rule for describing the ground state configurations. Taking iron ($_{26}\text{Fe}$) as an example, its neutral ground state configuration is measured as $[\text{Ar}] 4s^2 3d^6$, which agrees with Madelung’s rule. Now, consider $_{26}\text{Fe}^+$ which has one less electron, its ground state configuration would be predicted the same as manganese ($_{25}\text{Mn}$) $[\text{Ar}] 4s^2 3d^5$ by Madelung’s rule, which is, however, incorrect. The ground state configuration of $_{26}\text{Fe}^+$ is measured [24] as $[\text{Ar}] 4s^1 3d^6$, with one electron leaving from the $4s$ shell. This

5. Trends of many-body effects in atomic open-shells

Table 5.2.: Realistic ground state configurations of neutral atoms that differ from Madelung's rule.

Z	Element	Ground state configuration
24	Cr	[Ar] $4s^1 3d^5$
29	Cu	[Ar] $4s^1 3d^{10}$
41	Nb	[Kr] $5s^1 4d^4$
42	Mo	[Kr] $5s^1 4d^5$
44	Ru	[Kr] $5s^1 4d^7$
45	Rh	[Kr] $5s^1 4d^8$
46	Pd	[Kr] $4d^{10}$
47	Ag	[Kr] $5s^1 4d^{10}$
57	La	[Xe] $6s^2 5d^1$
58	Ce	[Xe] $6s^2 4f^1 5d^1$
64	Gd	[Xe] $6s^2 4f^7 5d^1$
78	Pt	[Xe] $6s^1 4f^{14} 5d^9$
79	Au	[Xe] $6s^1 4f^{14} 5d^{10}$
89	Ac	[Rn] $7s^2 6d^1$
90	Th	[Rn] $7s^2 6d^2$
91	Pa	[Rn] $7s^2 5f^2 6d^1$
92	U	[Rn] $7s^2 5f^3 6d^1$
93	Np	[Rn] $7s^2 5f^4 6d^1$
96	Cm	[Rn] $7s^2 5f^7 6d^1$
103	Lr	[Rn] $7s^2 5f^{14} 7p^1$

“ionization order” is observed (with exceptions, especially in f -blocks) that the electron with higher n leaves first (for the same n , higher l leaves first). To describe the general pattern of the ground state configurations of ionized atoms, we first apply the Aufbau principle with Madelung's rule to obtain the configuration of the neutral atom, then apply the “ionization order” to obtain the configurations of different oxidation states.

5.3. Trends of Slater-Condon parameters

The Slater-Condon parameters (or Slater integrals) [35, 36] are the radial components of the Coulomb matrix elements (4.12) obtained from the multipole expansion (4.11). From an experimental point of view, the Slater-Condon parameters are not directly measurable. They enter into atomic spectra via the multiplet calculations, that the multiplet energies are expressed in terms of combinations of $F^{(k)}$ (see Appendix B and [37, 38]). The task in this section is to investigate the periodic trends of the Slater-Condon parameters of

atomic open-shells. For open-shell systems with shell (n, l) , we integrate (5.1),

$$F_{nl}^{(k)} = \int_0^\infty dr_1 \int_0^\infty dr_2 \frac{r_1^k}{r_2^{k+1}} |u_{nl}(r_1)|^2 |u_{nl}(r_2)|^2 \quad (5.3)$$

First, we discuss a few analytical properties of this integral: 1. The k -dependence; 2. The Z -dependence for hydrogen-like wave functions; 3. How to integrate (5.3) in practice.

It can be seen from (5.3) that the integrand is everywhere positive, and since $\frac{r_1^0}{r_2^1} > \frac{r_1^1}{r_2^2} > \frac{r_1^2}{r_2^3} > \dots$ (a geometric sequence with common ratio less than 1), it follows that,

$$F^{(0)} > F^{(1)} > F^{(2)} > \dots > 0 \quad (5.4)$$

Being the radial parts of the Coulomb matrix elements, the Slater-Condon parameters have the dimension of an energy. Physically, the parameters with different k represent the Hartree energies of the electron densities expanded in multipoles [17]. The monopole term $F^{(0)}$ has the most significant contribution, and the values of higher multipole terms reduce for increasing k .

Multipole expansion of a charge density:

Consider a charge density of the form:

$$\rho(\mathbf{r}) = |R_{nl}(r)|^2 |Y_{lm}(\theta, \phi)|^2$$

Inserting the completeness relation of spherical harmonics, and using properties of Gaunt coefficients, we can express

$$\rho(\mathbf{r}) = \sum_{k=0}^{2l} \rho^{(k)}(\mathbf{r}) \quad (5.5)$$

where,

$$\rho^{(k)}(\mathbf{r}) = \langle lm | Y_{k0} | lm \rangle |R_{nl}(r)|^2 Y_{k0}(\theta, \phi) \quad (5.6)$$

is the k -th multipole term expanded from the charge density. Now, consider the Coulomb energy between two multipole charge densities:

$$E^{(k,k')} = \int d\mathbf{r} \int d\mathbf{r}' \frac{\rho^{(k)}(\mathbf{r}) \rho^{(k')}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (5.7)$$

Expanding $\frac{1}{|\mathbf{r} - \mathbf{r}'|}$, and using the orthonormalization relation of spherical harmonics,

$$\begin{aligned} E^{(k,k')} &= \langle lm | Y_{k0} | lm \rangle \langle lm | Y_{k'0} | lm \rangle \sum_{\lambda=0}^{\infty} F^{(\lambda)} \frac{4\pi}{2\lambda+1} \sum_{\mu=-\lambda}^{\lambda} \langle \lambda\mu | k0 \rangle \langle k'0 | \lambda\mu \rangle \\ &= \frac{4\pi \langle lm | Y_{k0} | lm \rangle^2}{2k+1} F^{(k)} \delta_{k,k'} \end{aligned} \quad (5.8)$$

5. Trends of many-body effects in atomic open-shells

we can see that the Coulomb energies associated with monopole-dipole, monopole-quadruple, dipole-quadruple, etc. interactions are *zero*.

The Hartree energy of the charge density is simply,

$$E_{\text{Hartree}} = \frac{1}{2} \sum_{k,k'} E^{(k,k')} = \frac{1}{2} \sum_{k=0}^{2l} E^{(k)} \quad (5.9)$$

where,

$$E^{(k)} = \frac{4\pi \langle lm | Y_{k0} | lm \rangle^2}{2k+1} F^{(k)} \quad (5.10)$$

is the Coulomb energy of the k -th multipole interacting with itself.

Consider hydrogen and hydrogen-like wave functions. Suppose,

$$F_{\text{H}}^{(k)} = \int_0^\infty dr_1 \int_0^\infty dr_2 \frac{r_{\leq}^k}{r_{>}^{k+1}} |u_{\text{H}}(r_1)|^2 |u_{\text{H}}(r_2)|^2 \quad (5.11)$$

are the integral values for hydrogen wave functions. From the rescaling relation (3.11), we can see that,

$$\begin{aligned} F_Z^{(k)} &= \int_0^\infty dr_1 \int_0^\infty dr_2 \frac{r_{\leq}^k}{r_{>}^{k+1}} |u_Z(r_1)|^2 |u_Z(r_2)|^2 \\ &= \int_0^\infty dr_1 \int_0^\infty dr_2 \frac{r_{\leq}^k}{r_{>}^{k+1}} |Z^{\frac{1}{2}} u_{\text{H}}(Zr_1)|^2 |Z^{\frac{1}{2}} u_{\text{H}}(Zr_2)|^2 \\ &= \frac{Z^{k+1}}{Z^k} \int_0^\infty d\rho_1 \int_0^\infty d\rho_2 \frac{\rho_{\leq}^k}{\rho_{>}^{k+1}} |u_{\text{H}}(\rho_1)|^2 |u_{\text{H}}(\rho_2)|^2 \\ &= Z F_{\text{H}}^{(k)} \end{aligned} \quad (5.12)$$

Therefore, the Slater-Condon parameters increase linearly with Z for hydrogen-like wave functions. The physical picture is that, as Z increases, the hydrogen-like wave functions shrink. The concentrated densities result in larger Hartree energies.

Now, we discuss how to integrate the Slater integrals in practice. The two-dimensional integral (5.3) can be reduced to one-dimensional integrals by considering the fact, that $\frac{r_{\leq}^k}{r_{>}^{k+1}}$ is either $\frac{r_2^k}{r_1^{k+1}}$ (when $r_1 > r_2$) or $\frac{r_1^k}{r_2^{k+1}}$ (when $r_1 < r_2$). Thus Eqn. (5.3) can be reduced to one-dimensional integrals piece-wise,

$$F_{nl}^{(k)} = \underbrace{\int_0^\infty dr_1 \frac{|u_{nl}(r_1)|^2}{r_1^{k+1}} \int_0^{r_1} dr_2 r_2^k |u_{nl}(r_2)|^2}_{r_1 > r_2} + \underbrace{\int_0^\infty dr_1 r_1^k |u_{nl}(r_1)|^2 \int_{r_1}^\infty dr_2 \frac{|u_{nl}(r_2)|^2}{r_2^{k+1}}}_{r_1 < r_2} \quad (5.13)$$

More instructively, we can evaluate (5.13) step-by-step as follows,

$$A(r) = \int_0^r dr' r'^k |u_{nl}(r')|^2 \quad (5.14)$$

$$B(r) = \int_r^\infty dr' \frac{|u_{nl}(r')|^2}{r'^{k+1}} \quad (5.15)$$

$$F_{nl}^{(k)} = \int_0^\infty dr |u_{nl}(r)|^2 \left[\frac{A(r)}{r^{k+1}} + r^k B(r) \right] \quad (5.16)$$

To be consistent with our earlier discussions, we compute the integrals numerically on the logarithmic grid (3.12). Recall that $dr = r dx$, the routine is implemented in Algorithm 5.1. Here we re-use the `cumsimps` function defined in Algorithm 3.6 to integrate (5.14) and (5.15) cumulatively.

Algorithm 5.1: Compute the Slater-Condon parameter $F_{nl}^{(k)}$ on a logarithmic grid

```

1 import scipy.integrate as sp
2
3 # Slater-Condon parameter
4 def Fk(u, k, r, dx):
5     A = cumsimps(u**2*r**(k+1), dx)
6     B = cumsimps((u**2/r**k)[::-1], dx)[::-1]
7     return sp.simps(u**2*(A/(r**k)+B*r**(k+1)), dx=dx)

```

Now, we are in a position to collect the SCF orbitals over the entire periodic table, and compute the corresponding Slater-Condon parameters. As we mentioned in the previous chapter, the angular part of the Coulomb matrix element (4.12) vanishes under certain combinations of the angular momenta. For a shell with angular number l , the non-trivial expansion indices are $k = 0, 2, 4, \dots, 2l$. In other words,

- for s -shells we calculate only $F^{(0)}$;
- for p -shells we calculate $F^{(0)}$ and $F^{(2)}$;
- for d -shells we calculate $F^{(0)}$, $F^{(2)}$, and $F^{(4)}$;
- for f -shells we calculate $F^{(0)}$, $F^{(2)}$, $F^{(4)}$, and $F^{(6)}$.

We first perform SCF calculations for all neutral elements with atomic number $Z \leq 118$: from hydrogen (${}_1\text{H}$) to Oganesson (${}_{118}\text{Og}$). From the SCF results, we compute the Slater-Condon parameters of the open-shells for each atom (the last shell filled according to the Aufbau principle). In the calculations, the Aufbau principle with Madelung's rule is used to set up the electronic configurations. Adopting the rules has the advantage to generate smooth data, so that the special cases (Table 5.2) do not distract our attention, and the overall periodic trends become clear to us.

The results are shown in Fig. 5.2. Different k values are plotted with different markers: $F^{(0)}$ circle; $F^{(2)}$ cross; $F^{(4)}$ square; $F^{(6)}$ hexagram. One can also observe that: for a given block, $F^{(k)}$ increases smoothly as a function of the atomic number Z . For example, from ${}_{21}\text{Sc}$ to ${}_{30}\text{Zn}$, $F^{(0)}$, $F^{(2)}$, and $F^{(4)}$ all increase as functions of Z . The physical picture is the following: When Z increases, the nuclear charge increases and the number of electrons also

5. Trends of many-body effects in atomic open-shells

increases. This additional electron, however, does not screen the nuclear charge effectively, because the electron is “on-shell” (in contrast to the inner shell electrons). Due to the larger nuclear attraction, the $3d$ wave functions shrink as Z increases (in general, within a block, the atomic radii shrink as Z increases [39]). These concentrated wave functions give results to larger Hartree energies and the corresponding larger values of $F^{(k)}$. On the other hand, when a new shell starts, the electron in the new shell experiences only a small amount of effective nuclear charge. Hence the wave function spreads out and the corresponding values of $F^{(k)}$ are small.

After studying the trends for the neutral atoms, we generalize the problem from atoms to their ions. In particular, we study atoms that “lose” electrons, i.e. cations, for the reason that the atomic systems with fewer electrons are guaranteed to have bound states. By *interpolating* the data from the atoms and cations, we should be able to approximate systems with more arbitrary electronic configurations. For the DFT calculations, we use the Aufbau principle with Madelung’s rule to set up the neutral configurations, and apply the “ionization order” (as explained in the previous section) to construct the oxidation states. Of course, this study involves thousands of calculations, as we perform DFT calculations for each oxidation state for each element. Fig. 5.3 shows the complete parameter results from hydrogen (${}_1\text{H}$) to Oganesson (${}_{118}\text{Og}$) with their corresponding oxidation states. To represent oxidation levels, we use color coding from dark blue (neutral) to dark red (highly ionized). Within the same block, for example from ${}_{21}\text{Sc}$ to ${}_{30}\text{Zn}$, the number of open-shell electrons increases with Z . Therefore there are more oxidation states for larger Z within a block, which leads to the “triangular shape” for each block. Notice that Fig. 5.3 includes Fig. 5.2: the data shown in Fig. 5.2 are the ones in Fig. 5.3 with ionization level = 0. Along the oxidation level, we can observe that $F^{(k)}$ increases smoothly. And the physical picture should be clear: when the electrons are removed, the shell of interest experiences larger effective nuclear charge, thus the wave functions shrink and the corresponding values of $F^{(k)}$ increase.

To interpolate the general trends of the data, we perform a least squares fitting. Two smooth trends are observed in the DFT calculations.

1. For a given block, $F^{(k)}$ increases smoothly with increasing atomic number.
2. For a given atom, $F^{(k)}$ increases smoothly with increasing oxidation level.

First we introduce a variable N_{auf} , which is the Aufbau filling number on the last shell. For instance, ${}_{26}\text{Fe}$ is the 6th element in the $3d$ -block, hence $N_{\text{auf}} = 6$; ${}_{47}\text{Ag}$ is the 9th element in the $4d$ -block, hence $N_{\text{auf}} = 9$; etc..

The first step is to fit the N_{auf} dependence for a given block and a fixed oxidation state. For instance, we consider the $3d$, $4d$, $5d$, $6d$ shells of neutral atoms. To a very good approximation, $F^{(k)}$ in each shell can be fitted perfectly by a parabola (Fig. 5.4a).

On the other hand, if we fix an atom and vary the oxidation level, we observe a piece-wise smooth trend. For instance, Fig. 5.4b plots $F^{(2)}$ of the open d -shell wave functions for Fe, Ru, Os, and Hs ions. The two different slopes are caused by taking electrons from different shells. Since the outer most s -shells have greater radii [40, 39], losing an electron

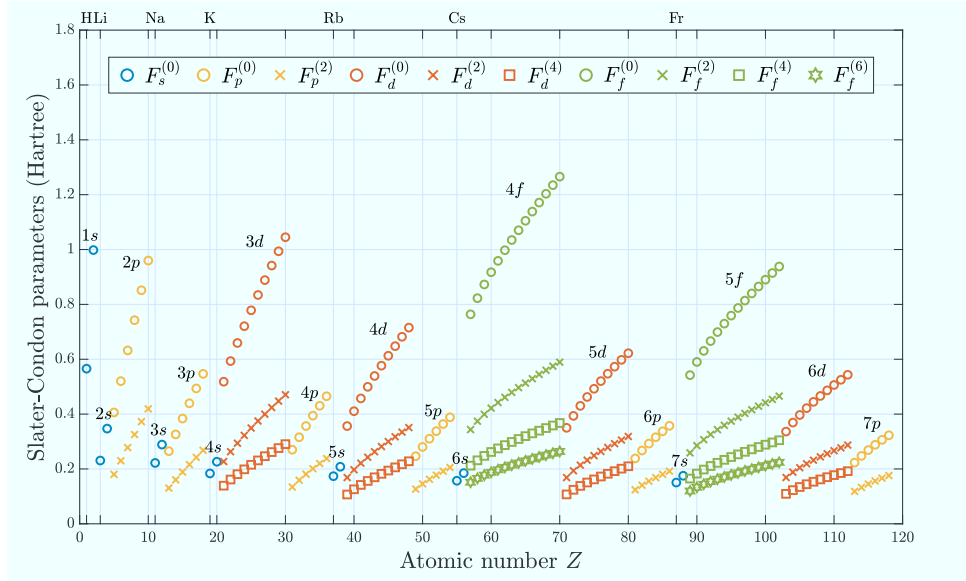


Figure 5.2.: Slater-Condon parameters calculated from LDA atomic open-shell orbitals for all neutral elements with atomic number $Z \leq 118$. The Aufbau principle with Madelung's rule is adopted for electronic configurations. (Number of data points: 318)

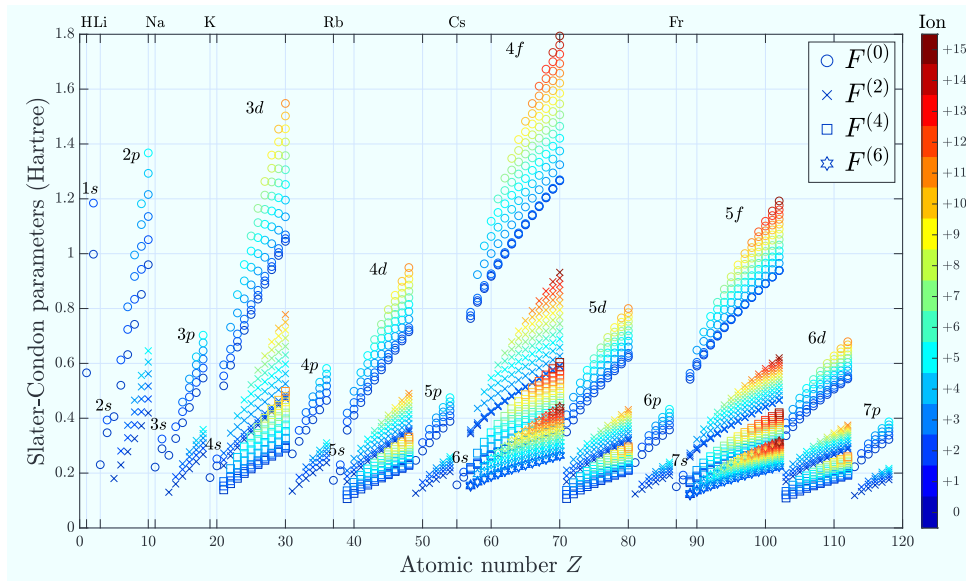


Figure 5.3.: Slater-Condon parameters calculated from LDA atomic open-shell orbitals for all elements with atomic number $Z \leq 118$ and their ions obtained by removing electrons from the open and outer shells. The Aufbau principle with Madelung's rule is adopted for electronic configurations. Colors from dark blue to dark red represent oxidation states: dark blue – neutral; dark red – highly ionized. (Number of data points: 2237)

5. Trends of many-body effects in atomic open-shells

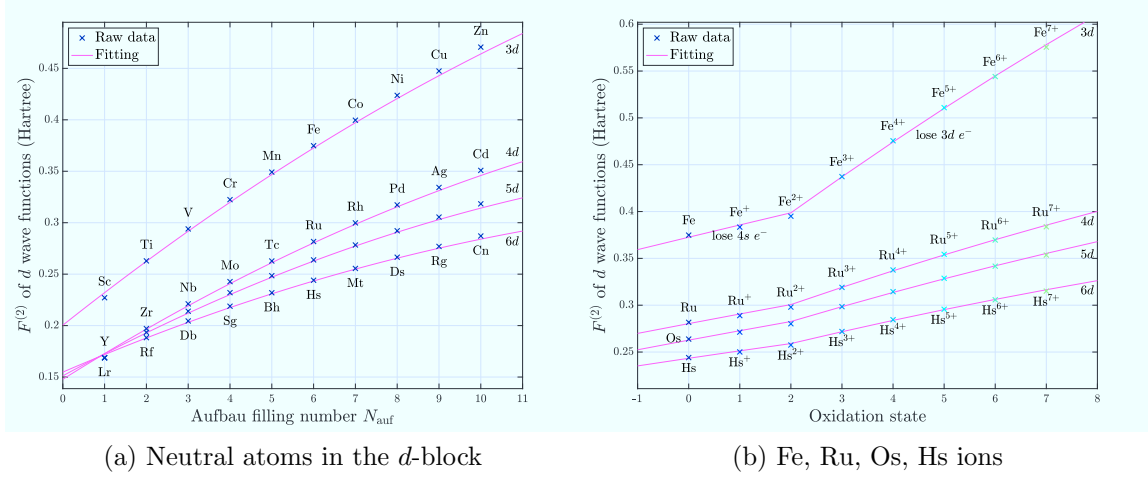


Figure 5.4.: Slater-Condon parameters $F^{(2)}$ for d -block atoms. (a) Trends of neutral atoms in $3d$, $4d$, $5d$, $6d$ blocks. Each shell can be well fitted by a parabola. (b) Trends of Fe, Ru, Os, Hs ions. Two slopes are observed for each atom due to two different sources of ionization. Fitting curves are generated from Table 5.4.

from the outer most s -shells has a much weaker influence on the open d -shells comparing with losing d -electrons itself.

In the previous section, we discussed the general pattern of the electron ionization order. In particular, for s - and p -blocks, the electrons start to leave directly from the open-shells; for d - and f -blocks, the electrons start to leave from the outer most s -shells. For example, ionizing a $3d$ shell requires first emptying the outer $4s$ -shell. For f -blocks, however, the realistic electronic configurations for the atoms and ions are rather irregular [24]. Here we simplify the pattern for f -blocks that we first empty the outer most s -shells, then directly remove electrons from the open-shells, which agrees with most of the experimental results (e.g. Eu, $\dots$, Eu $^{4+}$; Am, $\dots$, Am $^{6+}$; etc. [24]).

Taking into account the effect of removing electrons from different shells, we introduce a new variable q_{eff} , being the “effective charge”. If we assign the value of q_{eff} by losing an on-shell electron to be 1, then the value of q_{eff} by losing an outer-shell electron should have a value less than 1. For the d - and f -blocks, it is simply

$$q_{\text{eff}} = q_{\text{on-shell}} + \lambda q_{\text{outer-most-}s\text{-shell}} \quad (5.17)$$

For instance, if $\lambda = 0.33$, the q_{eff} values for $_{26}\text{Fe}$ ions are shown in Table 5.3.

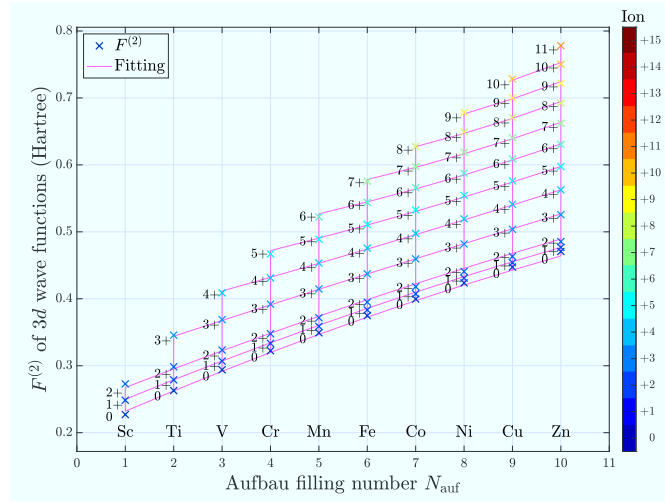
For each block, we have two variables: N_{auf} and q_{eff} . The fitting is given as,

$$F^{(k)}(N_{\text{auf}}, q_{\text{eff}}) = (\alpha_0 + \alpha_1 q_{\text{eff}} + \alpha_2 q_{\text{eff}}^2) + (\beta_0 + \beta_1 q_{\text{eff}}) N_{\text{auf}} + (\gamma_0 + \gamma_1 q_{\text{eff}}) N_{\text{auf}}^2 \quad (5.18)$$

where α_0 , α_1 , α_2 , β_0 , β_1 , γ_0 , γ_1 and λ (inside q_{eff}) are fitting parameters listed in Table 5.4. Notice that for s - and p -shells no λ is used since the electrons are ionized directly from the open-shells. The s - and p -shells use less parameters as the number of data points is smaller. An example of fitting $F^{(2)}$ in the $3d$ -block is shown in Fig. 5.5.

Table 5.3.: Effective charge q_{eff} .

Atom	Configuration	Oxidation	q_{eff}
Fe	[Ar] $3d^6 4s^2$	0	0.00
Fe ⁺	[Ar] $3d^6 4s^1$	1	0.33
Fe ²⁺	[Ar] $3d^6$	2	0.66
Fe ³⁺	[Ar] $3d^5$	3	1.66
Fe ⁴⁺	[Ar] $3d^4$	4	2.66
Fe ⁵⁺	[Ar] $3d^3$	5	3.66
Fe ⁶⁺	[Ar] $3d^2$	6	4.66
Fe ⁷⁺	[Ar] $3d^1$	7	5.66

Figure 5.5.: Least squares fitting of $F^{(2)}$ for the $3d$ -block atoms with different oxidation states. The fitting parameters are given in Table 5.4.

The fitting of the parameters is not merely an action of compressing the thousands of data points. It informs us the trends of the parameters, and we can interpolate systems with different electronic configurations. For instance, consider a copper atom $_{29}\text{Cu}$ with Madelung's configuration $[\text{Ar}] 4s^2 3d^9$. Setting ($N_{\text{auf}} = 9, q_{\text{eff}} = 0$) recovers the fitted data (in Hartree):

$$\begin{aligned} \text{Fit: } F^{(0)} &= 0.98 & F^{(2)} &= 0.44 & F^{(4)} &= 0.27 \\ \text{LDA: } F^{(0)} &= 0.99 & F^{(2)} &= 0.45 & F^{(4)} &= 0.28 \end{aligned}$$

Now, consider $_{29}\text{Cu}$ with the realistic configuration $[\text{Ar}] 4s^1 3d^{10}$. This configuration is not computed among the data that are used in the study of the trends. Comparing with Madelung's configuration, the $4s$ orbital loses one electron and the $3d$ orbital gains one electron. That corresponds to $q_{\text{eff}} = \lambda - 1 = -0.67$. Thus, setting ($N_{\text{auf}} = 9, q_{\text{eff}} = -0.67$), we obtain results for the realistic configuration (in Hartree):

$$\begin{aligned} \text{Fit: } F^{(0)} &= 0.94 & F^{(2)} &= 0.41 & F^{(4)} &= 0.25 \\ \text{LDA: } F^{(0)} &= 0.93 & F^{(2)} &= 0.41 & F^{(4)} &= 0.25 \end{aligned}$$

5. Trends of many-body effects in atomic open-shells

Table 5.4.: Fitting parameters for calculated open-shell Slater-Condon parameters.

Shell	k	α_0	α_1	α_2	β_0	β_1	γ_0	γ_1	λ
1s	0	0.13	0.19	—	0.43	—	—	—	—
2s	0	0.11	0.046	—	0.12	—	—	—	—
3s	0	0.15	0.036	—	0.067	—	—	—	—
4s	0	0.14	0.025	—	0.043	—	—	—	—
5s	0	0.14	0.022	—	0.034	—	—	—	—
6s	0	0.13	0.019	—	0.028	—	—	—	—
7s	0	0.13	0.017	—	0.025	—	—	—	—
2p	0	0.30	0.096	—	0.11	-2.5e-3	—	—	—
	2	0.13	0.053	—	0.048	-1.2e-3	—	—	—
3p	0	0.21	0.048	—	0.057	-3.0e-3	—	—	—
	2	0.10	0.029	—	0.028	-1.9e-3	—	—	—
4p	0	0.24	0.047	—	0.039	-4.1e-3	—	—	—
	2	0.12	0.030	—	0.021	-2.8e-3	—	—	—
5p	0	0.22	0.036	—	0.029	-3.4e-3	—	—	—
	2	0.11	0.024	—	0.016	-2.3e-3	—	—	—
6p	0	0.22	0.033	—	0.024	-3.2e-3	—	—	—
	2	0.11	0.022	—	0.014	-2.2e-3	—	—	—
7p	0	0.21	0.029	—	0.020	-2.8e-3	—	—	—
	2	0.11	0.020	—	0.012	-2.0e-3	—	—	—
3d	0	0.46	0.099	-1.0e-3	0.068	-8.4e-3	-1.1e-3	4.9e-4	0.33
	2	0.20	0.059	-6.3e-4	0.032	-4.9e-3	-5.9e-4	2.9e-4	
	4	0.12	0.039	-3.9e-4	0.020	-3.2e-3	-3.8e-4	1.9e-4	
4d	0	0.32	0.054	-2.9e-4	0.049	-6.0e-3	-9.8e-4	3.3e-4	0.53
	2	0.15	0.035	-3.6e-4	0.026	-3.9e-3	-5.7e-4	2.2e-4	
	4	0.093	0.025	-2.5e-4	0.017	-2.7e-3	-3.9e-4	1.5e-4	
5d	0	0.32	0.048	-2.7e-4	0.039	-5.9e-3	-8.9e-4	3.2e-4	0.60
	2	0.15	0.033	-3.3e-4	0.022	-4.0e-3	-5.7e-4	2.2e-4	
	4	0.095	0.024	-2.3e-4	0.015	-2.8e-3	-4.0e-4	1.5e-4	
4f	0	0.72	0.088	-1.0e-3	0.050	-6.0e-3	-8.9e-4	2.6e-4	0.10
	2	0.32	0.055	-6.4e-4	0.026	-3.5e-3	-5.1e-4	1.5e-4	
	4	0.20	0.037	-4.0e-4	0.016	-2.3e-3	-3.3e-4	9.7e-5	
	6	0.14	0.027	-2.9e-4	0.012	-1.7e-3	-2.4e-4	7.1e-5	
5f	0	0.51	0.048	-4.0e-4	0.040	-3.9e-3	-7.4e-4	1.6e-4	0.16
	2	0.24	0.031	-3.4e-4	0.022	-2.5e-3	-4.4e-4	1.1e-4	
	4	0.15	0.022	-2.4e-4	0.015	-1.7e-3	-3.1e-4	7.3e-5	
	6	0.11	0.017	-1.8e-4	0.011	-1.3e-3	-2.3e-4	5.5e-5	

5.4. The Hubbard U_{avg} and Hund's exchange J_{avg}

In general, a Coulomb matrix element (4.9)

$$\ddot{U}_{\alpha\beta\gamma\delta} = U_{\alpha\beta\gamma\delta} - U_{\alpha\beta\delta\gamma}$$

involves four orbital indices. The most important ones are the two-index (m_1, m_2) terms within a shell (n, l) (of the same spin) [41, 42, 43, 44, 45]:

$$\ddot{U}_{m_1 m_2} = U_{m_1 m_2} - J_{m_1 m_2} \quad (5.19)$$

where $U_{m_1 m_2}$ and $J_{m_1 m_2}$ are known as the direct and exchange Coulomb integrals, respectively. They are, (the form of the two-electron integrals is given in (1.18))

$$U_{m_1 m_2} = (m_1 m_2 | \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} | m_2 m_1) \quad (5.20)$$

$$J_{m_1 m_2} = (m_1 m_2 | \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} | m_1 m_2) \quad (5.21)$$

In practical calculations, it is useful to introduce the averaged quantities, where the angular indices m_1 and m_2 are averaged. For a given shell, there are $(2l+1)^2$ elements of $\ddot{U}_{m_1 m_2}$. But all the diagonal elements are zeros (since $U_{mm} - J_{mm} = 0$). Excluding the diagonals, there are $2l(2l+1)$ elements, whose average is,

$$\ddot{U}_{\text{avg}} = \frac{1}{2l(2l+1)} \sum_{m_1 m_2} \ddot{U}_{m_1 m_2} = \frac{1}{2l(2l+1)} \sum_{m_1 m_2} (U_{m_1 m_2} - J_{m_1 m_2}) \quad (5.22)$$

Similar to Eqn. (5.19), we also separate $\ddot{U}_{\text{avg}}$ into two terms:

$$\boxed{\ddot{U}_{\text{avg}} \equiv U_{\text{avg}} - J_{\text{avg}}} \quad (5.23)$$

which are known as the Hubbard U_{avg} and Hund's exchange J_{avg} , respectively. The Hubbard U_{avg} is defined as: [41, 42, 44, 45]

$$\boxed{U_{\text{avg}} \equiv \frac{1}{(2l+1)^2} \sum_{m_1 m_2} U_{m_1 m_2}} \quad (5.24)$$

There is no more degrees of freedom to define Hund's exchange J_{avg} . It just follows from (5.23) and (5.24): [41, 42, 44, 45]

$$J_{\text{avg}} = U_{\text{avg}} - \ddot{U}_{\text{avg}} \quad (5.25)$$

More explicitly, we can re-write J_{avg} as

$$J_{\text{avg}} = \frac{1}{2l(2l+1)} \sum_{m_1 m_2} J_{m_1 m_2} - \frac{1}{2l} U_{\text{avg}} \quad (5.26)$$

5. Trends of many-body effects in atomic open-shells

Explicitly expressing $U_{m_1 m_2}$ as (4.12) in terms of the Slater-Condon parameters $F^{(k)}$ and the Gaunt coefficients, plus some properties of Gaunt coefficients, we can show that U_{avg} is just $F^{(0)}$:

$$\begin{aligned}
 U_{\text{avg}} &= \frac{1}{(2l+1)^2} \sum_{k=0}^{2l} F^{(k)} \frac{4\pi}{2k+1} \sum_{m_1 m_2} \sum_{\mu=-k}^k \langle l m_1 | Y_{k\mu} | l m_1 \rangle \langle l m_2 | \overline{Y_{k\mu}} | l m_2 \rangle \\
 &= \frac{1}{(2l+1)^2} \sum_{k=0}^{2l} F^{(k)} \frac{4\pi}{2k+1} \left(\sum_m \langle l m | Y_{k0} | l m \rangle \right)^2 \\
 &= \frac{1}{(2l+1)^2} \sum_{k=0}^{2l} F^{(k)} \frac{4\pi}{2k+1} \left(\frac{(2l+1)^2}{4\pi} \delta_{k0} \right) \\
 &= F^{(0)}
 \end{aligned} \tag{5.27}$$

Similarly, we can show that J_{avg} are linear combinations of $F^{(k)}$:

$$\begin{aligned}
 J_{\text{avg}} &= \frac{1}{2l(2l+1)} \sum_{k=0}^{2l} F^{(k)} \frac{4\pi}{2k+1} \sum_{m_1 m_2} \sum_{\mu=-k}^k \langle l m_1 | Y_{k\mu} | l m_2 \rangle \langle l m_2 | \overline{Y_{k\mu}} | l m_1 \rangle - \frac{1}{2l} F^{(0)} \\
 &= \frac{1}{2l(2l+1)} \sum_{k=0}^{2l} F^{(k)} \frac{4\pi}{2k+1} \sum_{m_1 m_2} \langle l m_1 | Y_{k, m_1 - m_2} | l m_2 \rangle^2 - \frac{1}{2l} F^{(0)} \\
 &= \frac{1}{2l(2l+1)} \sum_{k=0}^{2l} F^{(k)} \frac{4\pi}{2k+1} \left((2l+1) \sqrt{\frac{2k+1}{4\pi}} \langle l 0 | Y_{k0} | l 0 \rangle \right) - \frac{1}{2l} F^{(0)} \\
 &= \frac{1}{2l} \sum_{k=0}^{2l} F^{(k)} \sqrt{\frac{4\pi}{2k+1}} \langle l 0 | Y_{k0} | l 0 \rangle - \frac{1}{2l} F^{(0)}
 \end{aligned} \tag{5.28}$$

In (5.28), the $F^{(0)}$ term vanishes, because $\langle l 0 | Y_{00} | l 0 \rangle = Y_{00} \langle l 0 | l 0 \rangle = Y_{00} = 1/\sqrt{4\pi}$. The final results reduce to linear combinations of $F^{(2)}, F^{(4)}, \dots, F^{(2l)}$ with Gaunt coefficients of the form $\langle l 0 | Y_{k0} | l 0 \rangle$. Inserting the analytical values of $\langle l 0 | Y_{k0} | l 0 \rangle$ (see Chapter 7), we summarize the results in Table 5.5.

Table 5.5.: Hubbard U_{avg} and Hund's exchange J_{avg} for s -, p -, d -, and f -shells.

s -shell	$U_{\text{avg}} = F^{(0)}$	$J_{\text{avg}} = 0$
p -shell	$U_{\text{avg}} = F^{(0)}$	$J_{\text{avg}} = \frac{1}{5} F^{(2)}$
d -shell	$U_{\text{avg}} = F^{(0)}$	$J_{\text{avg}} = \frac{1}{14} F^{(2)} + \frac{1}{14} F^{(4)}$
f -shell	$U_{\text{avg}} = F^{(0)}$	$J_{\text{avg}} = \frac{2}{45} F^{(2)} + \frac{1}{33} F^{(4)} + \frac{50}{1287} F^{(6)}$

5.4. The Hubbard U_{avg} and Hund's exchange J_{avg}

In practical calculations, it is often assumed that the *ratios* among $F^{(2)}$, $F^{(4)}$, and $F^{(6)}$ are constants [41, 42, 43, 46, 44, 45]. This assumption reduces the number of parameters in the calculations. Indeed, the parameters with different k should not be considered as independent variables. The physical picture is that when the radial wave functions expand or shrink, the values of the corresponding parameters should decrease or increase together. Empirically, for d -ions, the ratio $F^{(4)}/F^{(2)}$ is approximated to be $5/8 = 0.625$ [41, 42, 43, 46, 44]. For f -ions, the ratios $F^{(4)}/F^{(2)}$ and $F^{(6)}/F^{(2)}$ are estimated from the hydrogen-like $4f$ radial wave functions [41, 46], which are $451/675$ and $1001/2025$, respectively (see Appendix A).

It can be seen that the ratios are sometimes chosen with little guidance. We believe that it is valuable to study the parameters over different ions and provide simple but more reliable ratios according to the LDA calculations. The results are plotted in Fig. 5.6. For a fixed oxidation state, the ratios are found to a high degree constants within each block; For the varying oxidation states, the ratios change within 10%. To a very good approximation, one can regard the ratios within each block as constants. For each block, among different oxidation states, we estimate a simple fraction for the average ratio. The estimated ratios are plotted in Fig. 5.6 in purple lines and listed in Table 5.6. Analytic hydrogen-like results are also plotted in dashed black lines as references. One can observe that the realistic ratios are slightly smaller than the hydrogen-like ratios.

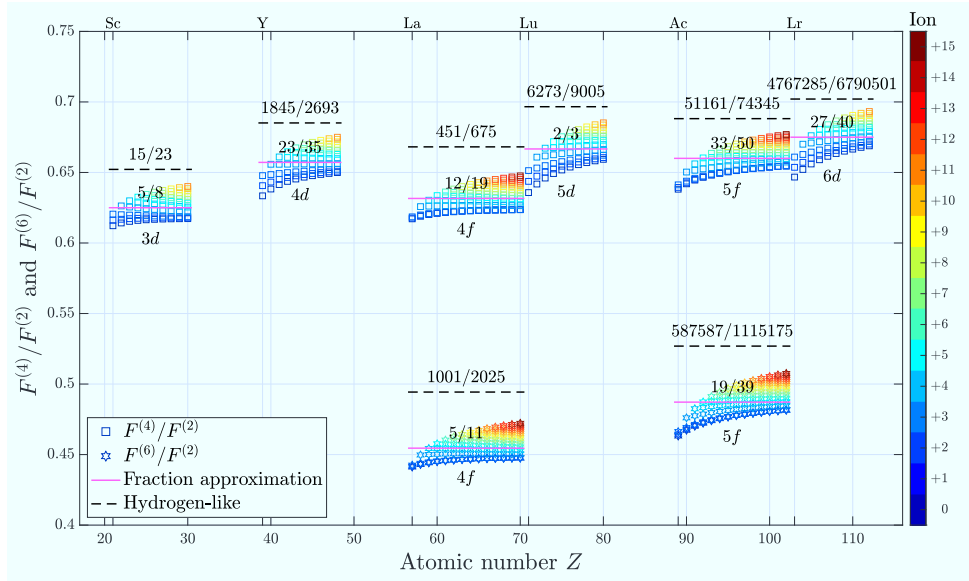


Figure 5.6.: Slater-Condon parameter ratios $F^{(4)}/F^{(2)}$ and $F^{(6)}/F^{(2)}$ for calculated d and f open-shells. The ratios are generated from the LDA data in Fig. 5.3. Simple fraction approximations around the averages for each block are provided. Hydrogen-like ratios are plotted in dashed black lines.

5. Trends of many-body effects in atomic open-shells

Table 5.6.: Simple fraction approximations of Slater-Condon parameter ratios $F^{(4)}/F^{(2)}$ and $F^{(6)}/F^{(2)}$ for calculated d and f open-shells.

Orbital	$F^{(4)}/F^{(2)}$	$F^{(6)}/F^{(2)}$
$3d$	$5/8$	—
$4d$	$23/35$	—
$5d$	$2/3$	—
$6d$	$27/40$	—
$4f$	$12/19$	$5/11$
$5f$	$33/50$	$19/39$

5.5. Trends of spin-orbit parameters

“The spin-orbit parameter Ξ_{nl} to H_{SO} ” is as “the Slater-Condon parameter $F_{nl}^{(k)}$ to H_U ”. They are both the radial components of the corresponding matrix elements (4.13) and (4.12), respectively. The spin-orbit parameters are directly related to atomic spectra measurements, in the sense that the spin-orbit term energies can be directly expressed as Ξ_{nl} multiplied by constants. The task in this section is to investigate the periodic trends of the spin-orbit parameters of atomic open-shells. For open-shell systems with shell (n, l) , we integrate (5.2),

$$\Xi_{nl} = \int_0^\infty dr |u_{nl}(r)|^2 \xi(r) \quad (5.29)$$

where, [21, 25]

$$\xi(r) = \frac{1}{2c^2 r} \frac{dV}{dr} \quad (5.30)$$

The speed of light is $c \approx 137.036 \text{ a}_0/t_0$ in atomic units; and $V(r)$ is the mean-field potential in the one-electron picture.

Being a relativistic effect, the spin-orbit parameter contains a factor of $1/c^2$, which implies that the spin-orbit interaction is normally weak. On the other hand, the spin-orbit parameter has a potential dependence (in contrast to the Slater-Condon parameter, which is only wave function dependent). If the potential has a large gradient (a steep potential), the spin-orbit interaction can be strong, which is normally the case for heavy elements.

Consider hydrogen and hydrogen-like wave functions and potentials, where,

$$\xi_Z(r) = \frac{Z}{2c^2 r^3} \quad (5.31)$$

Now, suppose

$$\Xi_{\text{H}} = \frac{1}{2c^2} \int_0^\infty dr \frac{|u_{\text{H}}(r)|^2}{r^3} \quad (5.32)$$

is the spin-orbit parameter for hydrogen wave functions and potential. From the rescaling relation (3.11), we can see that,

$$\Xi_Z = \frac{Z}{2c^2} \int_0^\infty dr \frac{|u_Z(r)|^2}{r^3} = \frac{Z}{2c^2} \int_0^\infty dr \frac{|Z^{\frac{1}{2}} u_{\text{H}}(Zr)|^2}{r^3} = \frac{Z^4}{2c^2} \int_0^\infty d\rho \frac{|u_{\text{H}}(\rho)|^2}{\rho^3} = Z^4 \Xi_{\text{H}} \quad (5.33)$$

Recall that $F_Z^{(k)}$ of the hydrogen-like systems increases linearly with Z , as shown in (5.12). Now, the spin-orbit parameter increases with Z to the power of 4, dramatic! Certainly, for realistic atoms, due to the screening effect, the scaling is less than the 4th power. But the strong Z -dependence can be still observed in the calculated results.

Another important aspect of the integral (5.29) is that it *diverges* for s -orbitals. For an atomic system, around the nucleus, we can approximately consider the mean-field potential as the pure nuclear potential. Thus $\xi(r) \propto 1/r^3$ as in (5.31). However, for s wave functions, we know that $u_s(r) \propto r$ when $r \rightarrow 0$, as given in (3.5). As a result, around 0, (5.29) integrates over $|u_s(r)|^2 \xi(r) \propto 1/r$, which diverges ($\Xi_s \rightarrow \infty$). But this doesn't mean that the spin-orbit interaction for s -orbitals is infinitely strong. Because on the other hand, the angular parts in (4.14) and (4.17) for s -orbitals are zeros. On physical grounds, there is no spin-orbit coupling when $l = 0$ (spin it there, but no orbit!). The spin-orbit parameters for s -shells are not well defined.

The routine for evaluating Ξ_{nl} is shown in Algorithm 5.2. It might be interesting to realize that the integral involves dr and the derivative involves $1/dr$. As a result, on either a uniform grid or a logarithmic grid, the dr or dx is canceled, thus not required as an input in the routine.

Algorithm 5.2: Compute the spin-orbit parameter Ξ_{nl}

```

1 import numpy as np
2 import scipy.integrate as sp
3
4 # Spin-orbit parameter
5 def Xi(u, V, r):
6     return sp.simps(u**2*np.gradient(V)/(2*137.036**2*r))

```

Again, we are in a position to collect the SCF orbitals over the entire periodic table, and compute the corresponding spin-orbit parameters. Similar to the study of the Slater-Condon parameters, we compute the spin-orbit parameters for all elements with their corresponding oxidation states for all $Z \leq 118$ (s -shells excluded). The results are plotted in Fig. 5.7.

Table 3.4 in Reference [25] provides spin-orbit parameters for a few atoms obtained from experimental data. Fig. 5.8 compares the spin-orbit parameters between the LDA calculations and the experimental results. We can see that both the values and the trends agree with each other to a high quality. For heavier elements, the reference data tend to have greater discrepancy to the calculated results. It should be pointed out that, the spin-orbit parameters from the experiments are obtained by using the LS -coupling scheme, which estimates the spin-orbit splitting in a perturbative approach. However, for heavy atoms, the energy gaps are effects of a mixture of both Coulomb and spin-orbit interactions. If one assumes that a certain energy gap is caused only by spin-orbit interaction, then the spin-orbit parameters will be over-estimated.

Comparing the Slater-Condon parameters in Fig. 5.3 with the spin-orbit parameters in Fig. 5.7, we see two main differences:

1. The $F^{(k)}$'s are of the order of $0.1 - 1$ Hartree. On the other hand, the Ξ 's are of the

5. Trends of many-body effects in atomic open-shells

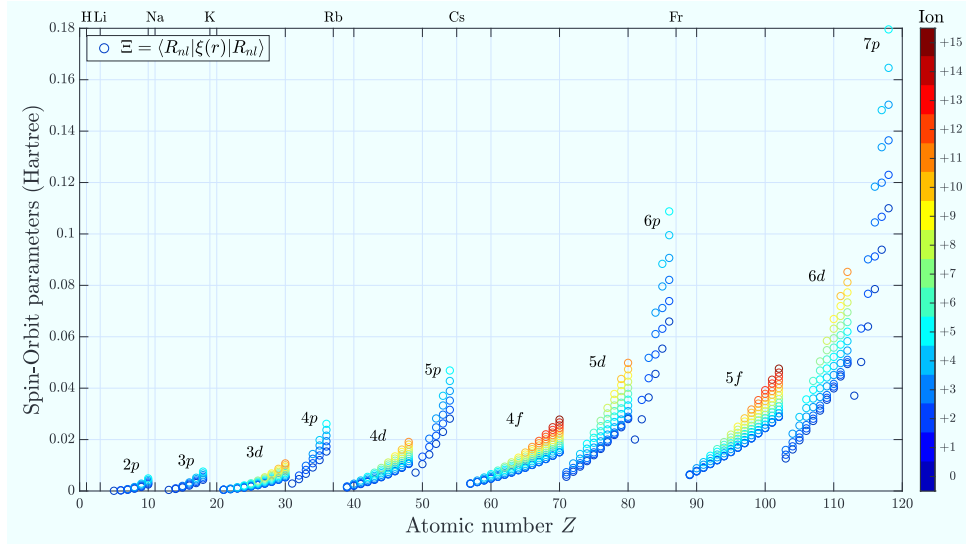


Figure 5.7.: Spin-orbit parameters calculated from LDA atomic open-shell orbitals for all elements with p , d , and f open-shells, with atomic number $Z \leq 118$ and their ions obtained by removing electrons from the open and outer shells. The Aufbau principle with Madelung's rule is adopted for electronic configurations. Colors from dark blue to dark red represent oxidation states: dark blue – neutral; dark red – highly ionized. (Number of data points: 692)

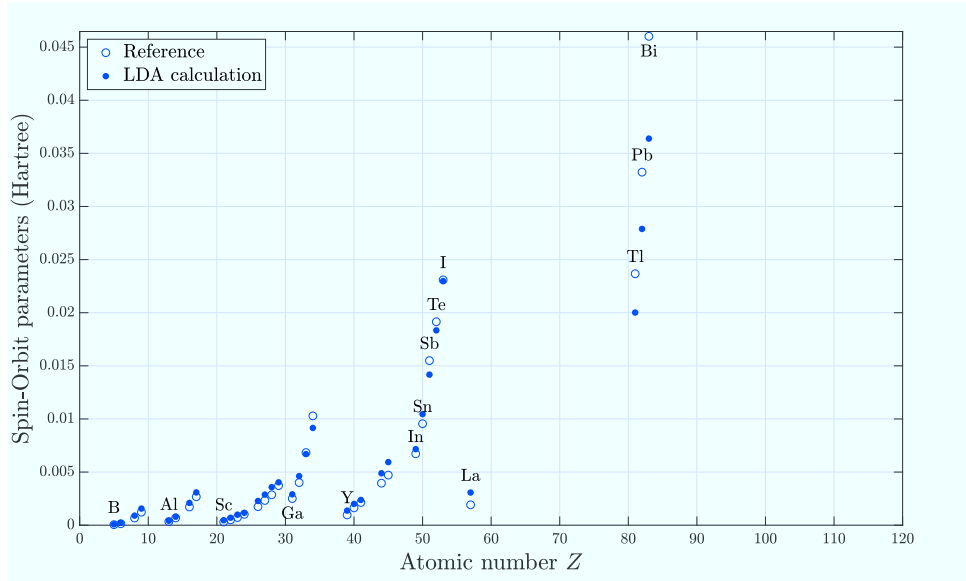


Figure 5.8.: Comparison of the spin-orbit parameters between experimental data and the LDA calculation results. Reference data obtained from Table 3.4 in [25].

order of 0.01 Hartree for light elements, and 0.1 Hartree for heavy elements.

2. Globally, the $F^{(k)}$'s do not have a strong Z -dependence. For a given k , the values are within the same order of magnitude over the entire periodic table. On the other

5.6. Trends of the Coulomb and spin-orbit interaction

hand, the Ξ 's show a strong Z -dependence, that the value increases dramatically for large Z .

Block-wise, the trend of the spin-orbit parameters are similar to the Slater-Condon parameters: For a given block, Ξ increases smoothly with increasing atomic number; for a given atom, Ξ increases smoothly with increasing oxidation level. Due to the similar block-wise trends, we perform the same fitting as in (5.18):

$$\Xi(N_{\text{auf}}, q_{\text{eff}}) = (\alpha_0 + \alpha_1 q_{\text{eff}} + \alpha_2 q_{\text{eff}}^2) + (\beta_0 + \beta_1 q_{\text{eff}}) N_{\text{auf}} + (\gamma_0 + \gamma_1 q_{\text{eff}}) N_{\text{auf}}^2 \quad (5.34)$$

where the fitting parameters are given in Table 5.7. By fitting the trends, one can estimate the parameters of systems with more arbitrary electronic configurations.

Table 5.7.: Fitting parameters for calculated open-shell spin-orbit parameters.

Shell	α_0	α_1	α_2	β_0	β_1	γ_0	γ_1	λ
$2p$	1.3e-4	2.1e-4	1.8e-5	-1.3e-4	-1.2e-4	8.4e-5	2.6e-5	—
$3p$	2.6e-4	2.6e-4	1.6e-5	5.9e-5	-3.8e-5	1.0e-4	1.5e-5	—
$4p$	1.6e-3	1.3e-3	5.2e-5	1.1e-3	2.0e-5	2.0e-4	1.4e-5	—
$5p$	4.3e-3	2.8e-3	8.0e-5	2.6e-3	-1.1e-5	2.3e-4	1.7e-5	—
$6p$	1.3e-2	7.6e-3	1.7e-4	6.8e-3	-1.7e-4	3.4e-4	3.3e-5	—
$7p$	2.5e-2	1.3e-2	2.6e-4	1.2e-2	-4.5e-4	3.8e-4	5.2e-5	—
$3d$	3.2e-4	1.5e-4	1.8e-5	1.0e-4	-1.3e-5	3.8e-5	3.9e-6	0.33
$4d$	9.0e-4	3.7e-4	3.0e-5	4.3e-4	-1.0e-5	5.1e-5	3.4e-6	0.53
$5d$	4.0e-3	1.4e-3	7.8e-5	1.5e-3	-3.6e-5	8.3e-5	4.7e-6	0.60
$6d$	9.6e-3	2.7e-3	1.1e-4	3.0e-3	-6.9e-5	9.4e-5	5.6e-6	0.59
$4f$	2.3e-3	5.3e-4	1.1e-5	4.9e-4	-8.4e-6	2.9e-5	2.1e-6	0.10
$5f$	5.0e-3	1.0e-3	1.4e-5	1.2e-3	-1.7e-5	3.5e-5	2.6e-6	0.16

5.6. Trends of the Coulomb and spin-orbit interaction

Consider a lead atom $_{82}\text{Pb}$ with $6p^2$ open-shell configuration. From the LDA calculations, one obtains the parameters (in Hartree):

$$F^{(0)} = 0.27; \quad F^{(2)} = 0.14; \quad \Xi = 0.028$$

Now, the question is: “Which interaction (Coulomb or spin-orbit) is relatively more important, in the sense that which coupling scheme (LS or jj) gives a better estimation for the energy spectra?”

The values of the parameters can misdirect us to believe that the Coulomb interaction is relatively more important. Because, apparently, the Slater-Condon parameters are one order of magnitude greater than the spin-orbit parameter. The suggestion is that the LS -coupling scheme is a better approximation scheme for the system, which is, however,

5. Trends of many-body effects in atomic open-shells

incorrect. Assessing the relative importance of Coulomb and spin-orbit interactions requires comparing the energy gaps among the spectra, rather than simply comparing the radial components of the matrix elements.

A quantitative and systematic estimation of the spectral gaps is the spectral *variance*. To compute the variance, a straightforward approach is to calculate the variance directly from the eigen-energies. This approach requires first diagonalizing the many-body Hamiltonians. In the next chapter, we will discuss a more powerful approach to compute the *moments* of general many-body Hamiltonians analytically without diagonalization. For the p^2 system, we know the eigen-energies, and hence the energy variances:

H_U		H_{SO}	
3P (9-fold)	$F^{(0)} - \frac{1}{5}F^{(2)}$	$(\frac{3}{2}, \frac{3}{2})$ (6-fold)	Ξ
1D (5-fold)	$F^{(0)} + \frac{1}{25}F^{(2)}$	$(\frac{3}{2}, \frac{1}{2})$ (8-fold)	$-\frac{1}{2}\Xi$
1S (1-fold)	$F^{(0)} + \frac{2}{5}F^{(2)}$	$(\frac{1}{2}, \frac{1}{2})$ (1-fold)	-2Ξ
$\text{Var} = \frac{18}{625}(F^{(2)})^2$		$\text{Var} = \frac{4}{5}\Xi^2$	

For the specific $6p^2$ open-shell of $_{82}\text{Pb}$, the values are (in Hartree²)

$$\text{Var}(H_U) = 0.00056 \quad \text{Var}(H_{SO}) = 0.00063$$

It turns out that the spectral variances are of the same order of magnitude. In this particular case, the jj -coupling scheme is marginally preferred than the LS -coupling scheme. But neither of the two are good approximation schemes, as the two interactions are of the same order of magnitude.

As an illustration, we compare the open-shell energy spectra for $_6\text{C}$, $_{14}\text{Si}$, $_{32}\text{Ge}$, $_{50}\text{Sn}$, $_{82}\text{Pb}$, and $_{114}\text{Fl}$. On the periodic table, they are in the same column (group 14) with p^2 open-shell configuration. The spectra are sketched in Fig. 5.9. In each diagram, LS -coupling results are shown on the left, and jj -coupling results are shown on the right. Referring to the intermediate-coupling results (middle column), we find that for $_6\text{C}$, $_{14}\text{Si}$, $_{32}\text{Ge}$, and $_{50}\text{Sn}$, the LS -coupling scheme is a suitable perturbative scheme, while on the other hand, the jj -coupling scheme does not produce reasonable results. For $_{82}\text{Pb}$, both LS -coupling and jj -coupling give similar estimations, as we have discussed. To see a spin-orbit dominated case, we move to the super heavy artificial element $_{114}\text{Fl}$, where the jj -coupling scheme becomes the preferred perturbative scheme.

Beyond p^2 , we study the trends of the spectral variances of the open-shell systems over the entire periodic table for all atoms with $Z \leq 118$. The explicit formulas for the spectral variances of different open-shell systems are listed in Table B.5. The final results are shown in Fig. 5.10. For open-shells with 1-electron or 1-hole, there is no Coulomb splitting but spin-orbit splitting is present. The spectral variances tend to be large when the open-shell is half-filled. In general, the Coulomb interaction is dominating until very heavy elements with $Z > 80$.

5.6. Trends of the Coulomb and spin-orbit interaction

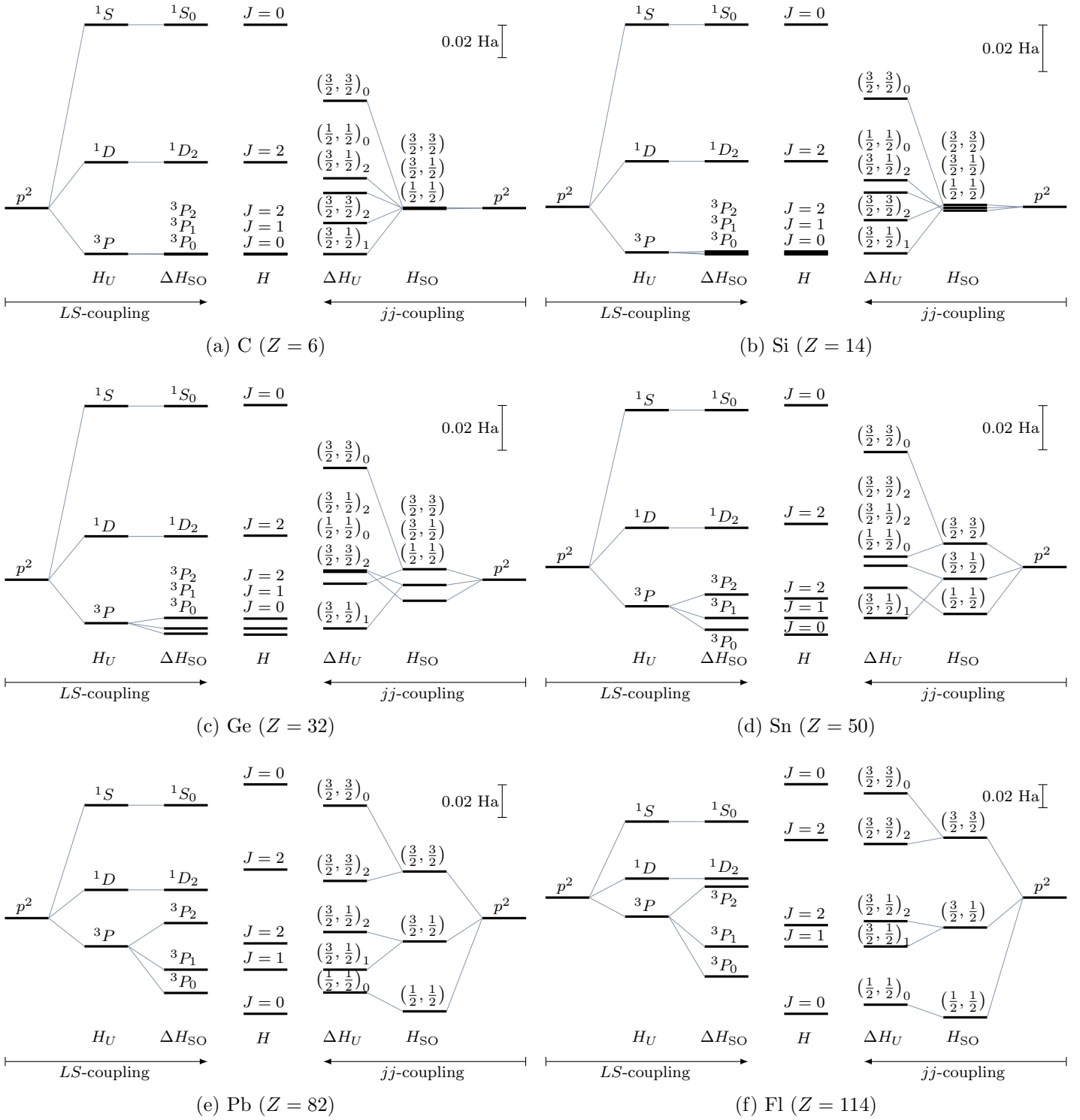


Figure 5.9.: Comparison between LS -coupling scheme and jj -coupling scheme for atoms ${}_6\text{C}$, ${}_{14}\text{Si}$, ${}_{32}\text{Ge}$, ${}_{50}\text{Sn}$, ${}_{82}\text{Pb}$, and ${}_{114}\text{Fl}$, with p^2 open-shell configuration. In each diagram: Left three columns: LS -coupling scheme; Right three columns: jj -coupling scheme; Middle column: Intermediate-coupling scheme.

5. Trends of many-body effects in atomic open-shells

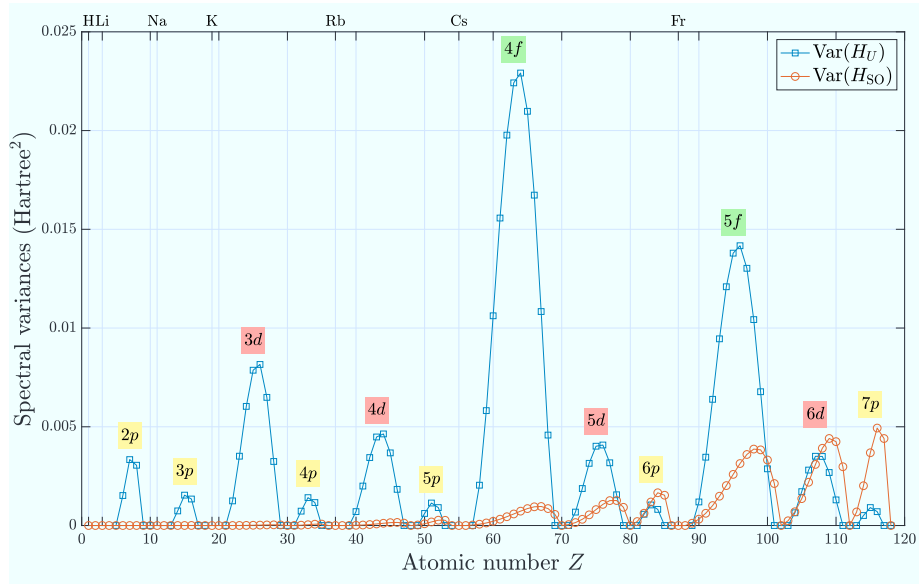


Figure 5.10.: Spectral variances of Coulomb and spin-orbit Hamiltonians of atomic open-shells for all atoms with atomic number $Z \leq 118$. The Aufbau principle with Madelung's rule is adopted for electronic configurations. For open-shells with 1-electron or 1-hole, there is no Coulomb splitting but spin-orbit splitting is present. The spectral variances tend to be large when the open-shell is half-filled. In general, the Coulomb interaction is dominating until very heavy elements with $Z > 80$.

6. Moments of many-electron Hamiltonians

6.1. Spectral moments

In the final part of Chapter 5, we introduced a problem of estimating the gaps of a given energy spectrum. The gaps are estimated by computing the spectral variances. If we have the complete set of the eigen-energies, we can easily compute the mean, the variance, and, in general, the n -th *moment* of the spectrum:

$$\langle E^n \rangle = \frac{1}{\dim_{\mathbf{H}}} \sum_{i=1}^{\dim_{\mathbf{H}}} E_i^n \quad (6.1)$$

Obviously, getting the eigen-energies of a many-electron Hamiltonian in the first place is the ultimate difficult task. So, we ask,

Can we evaluate $\langle E^n \rangle$ without having the eigen-energies?

Yes. The key is to realize that the *trace* of a Hamiltonian is independent of basis transformations. Suppose we have a matrix representation $\mathbf{H}$ in a many-electron basis. In general, the many-electron basis is not an eigen-basis, so $\mathbf{H}$ is not diagonal. But since the trace is invariant under basis transformations, there is no need to go to the eigen-basis:

$$\langle E^n \rangle = \frac{1}{\dim_{\mathbf{H}}} \text{Tr}(\mathbf{H}^n) \quad (6.2)$$

So, given a matrix representation, take the matrix to the power of n , average the diagonal elements, we get the n -th moment, and there is no matrix diagonalization involved. However, one does not gain much here as matrix multiplications and diagonalizations are of the same order of computational complexity ($\mathcal{O}(\dim_{\mathbf{H}}^3)$). The fundamental challenge is that the Hilbert space dimension that we are working with is in general enormous (recall Eqn. (1.11) and Table 1.1).

Consider general one- and two-body Hamiltonians,

$$H_1 = \sum_{\alpha\beta} h_{\alpha\beta} c_{\alpha}^{\dagger} c_{\beta} \quad \text{and} \quad H_2 = \sum_{\substack{\alpha<\beta \\ \gamma<\delta}} \ddot{v}_{\alpha\beta\gamma\delta} c_{\alpha}^{\dagger} c_{\beta}^{\dagger} c_{\gamma} c_{\delta} \quad (6.3)$$

In this chapter, we ask an ambitious question,

Can we evaluate $\langle E^n \rangle$ without working with the many-electron basis?

6.2. Moments of one-body Hamiltonians

We start from investigating the one-body Hamiltonian,

$$H = \sum_{\alpha\beta} h_{\alpha\beta} c_{\alpha}^{\dagger} c_{\beta} \quad (6.4)$$

In fact, if a system is described purely by a one-body Hamiltonian, the system is *non-interacting*. The many-electron problems reduce to one-electron problems. There is no need to work with a many-electron basis. However, we insist to discuss the moments of one-body Hamiltonians from a many-electron perspective. Hopefully it will give us an algebraic insight for later handling the two-body Hamiltonians.

6.2.1. The first moment

In general, in a many-electron basis, a matrix element of (6.4) is

$$\langle i | H | j \rangle = \sum_{\alpha\beta} h_{\alpha\beta} \langle i | c_{\alpha}^{\dagger} c_{\beta} | j \rangle \quad (6.5)$$

where $\langle i |$ and $| j \rangle$ are the many-electron basis states.

First, we consider the first moment of the spectrum (which is the spectral mean). To get the first moment, we average over all the diagonal elements:

$$\langle E \rangle = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{\alpha\beta} h_{\alpha\beta} \langle i | c_{\alpha}^{\dagger} c_{\beta} | i \rangle \quad (6.6)$$

Obviously,

$$\langle i | c_{\alpha}^{\dagger} c_{\beta} | i \rangle = \begin{cases} 1 & \text{if } (\alpha = \beta) \text{ and } (\alpha, \beta \in i) \\ 0 & \text{otherwise} \end{cases} \quad (6.7)$$

Hence, (6.6) reduces to

$$\langle E \rangle = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{\alpha \in i} h_{\alpha\alpha} \quad (6.8)$$

where $(\alpha \in i)$ denotes that α is an occupied orbital of basis state i . Now, within the complete set of basis states ($\dim_{\text{H}} = N_{\text{orb}} - \text{choose} - N_e$), how many basis states fulfill the condition that the orbital α is occupied? This is a conditional combinatorics problem: we “lock” one electron and one orbital to count the rest. There are $(N_{\text{orb}} - 1) - \text{choose} - (N_e - 1)$ possibilities. Thus, the sum in (6.8) simplifies to,

$$\sum_{i=1}^{\dim_{\text{H}}} \sum_{\alpha \in i} = \binom{N_{\text{orb}} - 1}{N_e - 1} \sum_{\alpha=1}^{N_{\text{orb}}} \quad (6.9)$$

The summation over the Hilbert space dimension disappears! Eqn. (6.9) is the *key* of collapsing a Hilbert space summation to a one-electron space summation. Further, we use a more compact notation,

$$\sum_{\alpha=1}^{N_{\text{orb}}} h_{\alpha\alpha} = \text{Tr}(\mathbf{h}) \quad (6.10)$$

where $\mathbf{h}$ is the matrix representation of the Hamiltonian in the one-electron basis. Finally, the first moment simplifies to,

$$\langle E \rangle = \frac{1}{\text{dim}_{\text{H}}} \binom{N_{\text{orb}}-1}{N_e-1} \text{Tr}(\mathbf{h}) \quad (6.11)$$

The original “ $\mathbf{H}$ -problem” ($\text{dim}_{\text{H}} = N_{\text{orb}}$ -choose- N_e) reduces to an “ $\mathbf{h}$ -problem” ($\text{dim}_{\text{h}} = N_{\text{orb}}$). If we denote

$$\langle \varepsilon^n \rangle = \frac{1}{\text{dim}_{\text{h}}} \text{Tr}(\mathbf{h}^n) \quad (6.12)$$

Eqn. (6.11) can be equivalently written as,

$$\langle E \rangle = N_e \langle \varepsilon \rangle \quad (6.13)$$

The mean eigen-energy of the many-electron system is the mean eigen-energy of the one-electron system times the number of electrons.

6.2.2. The second moment

Next, we consider the Hamiltonian squared: (note $H = H^\dagger$ is hermitian)

$$H^2 = \sum_{\alpha\beta\gamma\delta} h_{\alpha\beta} h_{\gamma\delta} c_\alpha^\dagger c_\beta c_\gamma^\dagger c_\delta \quad (6.14)$$

For the second moment, we trace the squared operator,

$$\langle E^2 \rangle = \frac{1}{\text{dim}_{\text{H}}} \sum_{i=1}^{\text{dim}_{\text{H}}} \sum_{\alpha\beta\gamma\delta} h_{\alpha\beta} h_{\gamma\delta} \langle i | c_\alpha^\dagger c_\beta c_\gamma^\dagger c_\delta | i \rangle \quad (6.15)$$

To work with the term $\langle i | c_\alpha^\dagger c_\beta c_\gamma^\dagger c_\delta | i \rangle$, we re-order the operators:

$$\langle i | c_\alpha^\dagger c_\beta c_\gamma^\dagger c_\delta | i \rangle = \langle i | c_\alpha^\dagger (\delta_{\beta\gamma} - c_\gamma^\dagger c_\beta) c_\delta | i \rangle = \langle i | c_\alpha^\dagger c_\delta | i \rangle \delta_{\beta\gamma} - \langle i | c_\alpha^\dagger c_\gamma^\dagger c_\beta c_\delta | i \rangle \quad (6.16)$$

Now,

$$\langle i | c_\alpha^\dagger c_\delta | i \rangle = \begin{cases} 1 & \text{if } (\alpha = \delta) \text{ and } (\alpha, \delta \in i) \\ 0 & \text{otherwise} \end{cases} \quad (6.17)$$

$$\langle i | c_\alpha^\dagger c_\gamma^\dagger c_\beta c_\delta | i \rangle = \begin{cases} 1 & \text{if } (\alpha = \delta) \text{ and } (\beta = \gamma) \text{ and } (\alpha, \beta, \gamma, \delta \in i) \\ -1 & \text{if } (\alpha = \beta) \text{ and } (\gamma = \delta) \text{ and } (\alpha, \beta, \gamma, \delta \in i) \\ 0 & \text{otherwise} \end{cases} \quad (6.18)$$

6. Moments of many-electron Hamiltonians

In a compact notation, we write,

$$\langle i | c_\alpha^\dagger c_\beta c_\gamma^\dagger c_\delta | i \rangle = \delta_{\alpha\delta} \delta_{\beta\gamma} \Delta(\alpha\delta \in i) - (\delta_{\alpha\delta} \delta_{\beta\gamma} - \delta_{\alpha\beta} \delta_{\gamma\delta}) \Delta(\alpha\beta\gamma\delta \in i) \quad (6.19)$$

where $\Delta(\text{condition})$ returns 1 if the condition is true, otherwise 0.

Substitute (6.19) back to (6.15), we obtain,

$$\langle E^2 \rangle = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \left(\sum_{\alpha \in i} \sum_{\beta} h_{\alpha\beta} h_{\beta\alpha} - \sum_{\alpha\beta \in i} h_{\alpha\beta} h_{\beta\alpha} + \sum_{\alpha\gamma \in i} h_{\alpha\alpha} h_{\gamma\gamma} \right) \quad (6.20)$$

Using the conditional combinatorics trick,

$$\langle E^2 \rangle = \frac{1}{\dim_{\text{H}}} \left[\binom{N_{\text{orb}}-1}{N_e-1} \sum_{\alpha\beta} h_{\alpha\beta} h_{\beta\alpha} - \binom{N_{\text{orb}}-2}{N_e-2} \sum_{\alpha\beta} h_{\alpha\beta} h_{\beta\alpha} + \binom{N_{\text{orb}}-2}{N_e-2} \sum_{\alpha\gamma} h_{\alpha\alpha} h_{\gamma\gamma} \right]$$

Combine the first two terms and collapse the sums into traces, we obtain the second moment in terms of the one-electron matrix $\mathbf{h}$:

$$\langle E^2 \rangle = \frac{1}{\dim_{\text{H}}} \left[\binom{N_{\text{orb}}-2}{N_e-1} \text{Tr}(\mathbf{h}^2) + \binom{N_{\text{orb}}-2}{N_e-2} \text{Tr}^2(\mathbf{h}) \right] \quad (6.21)$$

Equivalently,

$$\langle E^2 \rangle = \frac{N_e(N_{\text{orb}}-N_e)}{N_{\text{orb}}-1} \langle \varepsilon^2 \rangle + \frac{N_e(N_e-1)N_{\text{orb}}}{N_{\text{orb}}-1} \langle \varepsilon \rangle^2 \quad (6.22)$$

From (6.13) and (6.22), we can easily derive the many-electron spectral variance of a one-body Hamiltonian:

$$\text{Var}(E) = \langle E^2 \rangle - \langle E \rangle^2 = \frac{N_e(N_{\text{orb}}-N_e)}{N_{\text{orb}}-1} \left[\langle \varepsilon^2 \rangle - \langle \varepsilon \rangle^2 \right] \quad (6.23)$$

Consider a specific example. Suppose we have a many-electron atomic shell system with the spin-orbit Hamiltonian. For convenience, we use the jj -basis as the one-electron basis, since the one-electron matrix elements (4.53) are diagonal:

$$h_{\alpha\beta} = \frac{1}{2} \Xi_{nl} \left[j_\alpha(j_\alpha + 1) - l(l+1) - \frac{3}{4} \right] \delta_{\alpha\beta} \quad (6.24)$$

Given an atomic shell in the jj -basis (e.g. Fig. 4.1b), there are $(2l+2)$ orbitals with $j = l + \frac{1}{2}$, and $2l$ orbitals with $j = l - \frac{1}{2}$. It turns out that the trace of the one-electron spin-orbit matrix elements is zero,

$$\begin{aligned} \text{Tr}(\mathbf{h}) &= \frac{1}{2} \Xi_{nl} (2l+2) \left[\left(l + \frac{1}{2} \right) \left(l + \frac{3}{2} \right) - l(l+1) - \frac{3}{4} \right] \\ &\quad + \frac{1}{2} \Xi_{nl} (2l) \left[\left(l - \frac{1}{2} \right) \left(l + \frac{1}{2} \right) - l(l+1) - \frac{3}{4} \right] = 0 \end{aligned} \quad (6.25)$$

Since (6.24) is diagonal, $\mathbf{h}^2$ is simply a diagonal matrix squared. Therefore,

$$\begin{aligned}\text{Tr}(\mathbf{h}^2) &= \frac{1}{4}\Xi_{nl}^2(2l+2)\left[\left(l+\frac{1}{2}\right)\left(l+\frac{3}{2}\right)-l(l+1)-\frac{3}{4}\right]^2 \\ &\quad + \frac{1}{4}\Xi_{nl}^2(2l)\left[\left(l-\frac{1}{2}\right)\left(l+\frac{1}{2}\right)-l(l+1)-\frac{3}{4}\right]^2 \\ &= \frac{1}{4}\Xi_{nl}^2(4l+2)l(l+1)\end{aligned}\quad (6.26)$$

For an atomic shell, $\dim_{\mathbf{h}} = N_{\text{orb}} = 4l + 2$. Hence,

$$\langle \varepsilon \rangle = 0 \quad \text{and} \quad \langle \varepsilon^2 \rangle = \frac{l(l+1)}{4}\Xi_{nl}^2 \quad (6.27)$$

Therefore, the spectral variance of the spin-orbit Hamiltonian of a many-electron atomic shell system reads,

$$\boxed{\text{Var}(E) = \frac{N_e(N_{\text{orb}} - N_e)}{N_{\text{orb}} - 1} \frac{l(l+1)}{4} \Xi_{nl}^2} \quad (6.28)$$

6.2.3. The third moment

From the experience of working with the 2nd moment, we see that a crucial step is to evaluate the diagonal elements of a “properly ordered” (all creators left, all annihilators right) n -body operator (called the *density matrix*). In general, the diagonal elements of a density matrix can be expressed as a determinant of deltas [47]

$$\langle i | c_{\alpha_1}^\dagger c_{\alpha_2}^\dagger \cdots c_{\alpha_n}^\dagger c_{\beta_n} \cdots c_{\beta_2} c_{\beta_1} | i \rangle = \begin{vmatrix} \delta_{\alpha_1\beta_1} & \delta_{\alpha_1\beta_2} & \cdots & \delta_{\alpha_1\beta_n} \\ \delta_{\alpha_2\beta_1} & \delta_{\alpha_2\beta_2} & \cdots & \delta_{\alpha_2\beta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{\alpha_n\beta_1} & \delta_{\alpha_n\beta_2} & \cdots & \delta_{\alpha_n\beta_n} \end{vmatrix} \Delta(\alpha_1 \cdots \alpha_n \beta_n \cdots \beta_1 \in i) \quad (6.29)$$

Now, we consider the Hamiltonian cubed:

$$H^3 = \sum_{\alpha\beta\gamma\delta\epsilon\zeta} h_{\alpha\beta} h_{\gamma\delta} h_{\epsilon\zeta} c_\alpha^\dagger c_\beta c_\gamma^\dagger c_\delta c_\epsilon^\dagger c_\zeta \quad (6.30)$$

Here, for a better readability (and easier typing...), we re-write the above equation as,

$$H^3 = \sum_{ABCDEF} [AB][CD][EF] A^\dagger B C^\dagger D E^\dagger F \quad (6.31)$$

To evaluate $\langle E^3 \rangle$, we re-order the operators:

$$\begin{aligned}A^\dagger B C^\dagger D E^\dagger F &= A^\dagger (\delta_{BC} - C^\dagger B) (\delta_{DE} - E^\dagger D) F \\ &= A^\dagger (\delta_{BC} \delta_{DE} - \delta_{BC} E^\dagger D - C^\dagger B \delta_{DE} + C^\dagger B E^\dagger D) F \\ &= A^\dagger (\delta_{BC} \delta_{DE} - \delta_{BC} E^\dagger D - C^\dagger B \delta_{DE} + C^\dagger (\delta_{BE} - E^\dagger B) D) F \\ &= A^\dagger F \delta_{BC} \delta_{DE} - A^\dagger E^\dagger D F \delta_{BC} - A^\dagger C^\dagger B F \delta_{DE} + A^\dagger C^\dagger D F \delta_{BE} - A^\dagger C^\dagger E^\dagger B D F\end{aligned}$$

6. Moments of many-electron Hamiltonians

Now, we trace for the 3rd moment,

$$\begin{aligned} \langle E^3 \rangle = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{ABCDEF} [AB][CD][EF] (& \\ & \langle i | A^\dagger F | i \rangle \delta_{BC} \delta_{DE} \\ & - \langle i | A^\dagger E^\dagger DF | i \rangle \delta_{BC} \\ & - \langle i | A^\dagger C^\dagger BF | i \rangle \delta_{DE} \\ & + \langle i | A^\dagger C^\dagger DF | i \rangle \delta_{BE} \\ & - \langle i | A^\dagger C^\dagger E^\dagger BDF | i \rangle \\ &) \end{aligned} \quad (6.32)$$

Using (6.29), we can express each diagonal density matrix element as determinant of deltas. This won't look very pleasant, but,

$$\begin{aligned} \langle E^3 \rangle = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{ABCDEF} [AB][CD][EF] (& \\ & \delta_{AF} \Delta(AF \in i) \delta_{BC} \delta_{DE} \\ & - (\delta_{AF} \delta_{DE} - \delta_{AD} \delta_{EF}) \Delta(ADEF \in i) \delta_{BC} \\ & - (\delta_{AF} \delta_{BC} - \delta_{AB} \delta_{CF}) \Delta(ABCF \in i) \delta_{DE} \\ & + (\delta_{AF} \delta_{DC} - \delta_{AD} \delta_{CF}) \Delta(ACDF \in i) \delta_{BE} \\ & - (\delta_{AF} \delta_{CD} \delta_{BE} + \delta_{AD} \delta_{BC} \delta_{EF} + \delta_{AB} \delta_{CF} \delta_{DE} \\ & - \delta_{AB} \delta_{CD} \delta_{EF} - \delta_{AD} \delta_{CF} \delta_{BE} - \delta_{AF} \delta_{BC} \delta_{DE}) \Delta(ABCDEF \in i) \\ &) \\ = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} (& \\ & \sum_{A \in i} \sum_{BD} [AB][BD][DA] \\ & - \sum_{AD \in i} \sum_B [AB][BD][DA] + \sum_{AE \in i} \sum_B [AB][BA][EE] \\ & - \sum_{AB \in i} \sum_D [AB][BD][DA] + \sum_{AC \in i} \sum_D [AA][CD][DC] \\ & + \sum_{AC \in i} \sum_B [AB][CC][BA] - \sum_{AC \in i} \sum_B [AB][CA][BC] \\ & - \sum_{ABC \in i} [AB][CC][BA] - \sum_{ABE \in i} [AB][BA][EE] - \sum_{ACD \in i} [AA][CD][DC] \\ & + \sum_{ACE \in i} [AA][CC][EE] + \sum_{ABC \in i} [AB][CA][BC] + \sum_{ACD \in i} [AB][BD][DA] \\ &) \end{aligned} \quad (6.33)$$

Using the conditional combinatorics trick and expressing the sums as traces,

$$\begin{aligned}
\langle E^3 \rangle = \frac{1}{\dim_{\mathbf{H}}} & \left[\begin{aligned} & \binom{N_{\text{orb}}-1}{N_e-1} \text{Tr}(\mathbf{h}^3) \\ & - \binom{N_{\text{orb}}-2}{N_e-2} \text{Tr}(\mathbf{h}^3) + \binom{N_{\text{orb}}-2}{N_e-2} \text{Tr}(\mathbf{h}^2) \text{Tr}(\mathbf{h}) \\ & - \binom{N_{\text{orb}}-2}{N_e-2} \text{Tr}(\mathbf{h}^3) + \binom{N_{\text{orb}}-2}{N_e-2} \text{Tr}(\mathbf{h}^2) \text{Tr}(\mathbf{h}) \\ & + \binom{N_{\text{orb}}-2}{N_e-2} \text{Tr}(\mathbf{h}^2) \text{Tr}(\mathbf{h}) - \binom{N_{\text{orb}}-2}{N_e-2} \text{Tr}(\mathbf{h}^3) \\ & - \binom{N_{\text{orb}}-3}{N_e-3} \text{Tr}(\mathbf{h}^2) \text{Tr}(\mathbf{h}) - \binom{N_{\text{orb}}-3}{N_e-3} \text{Tr}(\mathbf{h}^2) \text{Tr}(\mathbf{h}) - \binom{N_{\text{orb}}-3}{N_e-3} \text{Tr}(\mathbf{h}^2) \text{Tr}(\mathbf{h}) \\ & + \binom{N_{\text{orb}}-3}{N_e-3} \text{Tr}^3(\mathbf{h}) + \binom{N_{\text{orb}}-3}{N_e-3} \text{Tr}(\mathbf{h}^3) + \binom{N_{\text{orb}}-3}{N_e-3} \text{Tr}(\mathbf{h}^3) \end{aligned} \right] \quad (6.34)
\end{aligned}$$

which reduces to,

$$\langle E^3 \rangle = \frac{1}{\dim_{\mathbf{H}}} \left[\left[\binom{N_{\text{orb}}-3}{N_e-1} - \binom{N_{\text{orb}}-3}{N_e-2} \right] \text{Tr}(\mathbf{h}^3) + 3 \binom{N_{\text{orb}}-3}{N_e-2} \text{Tr}(\mathbf{h}^2) \text{Tr}(\mathbf{h}) + \binom{N_{\text{orb}}-3}{N_e-3} \text{Tr}^3(\mathbf{h}) \right] \quad (6.35)$$

Algorithm 6.1 gives example codes for computing the spectral moments $\langle E^1 \rangle$, $\langle E^2 \rangle$, and $\langle E^3 \rangle$ of a one-body Hamiltonian for a many-electron system. Note that the input arguments of the functions are identical to the arguments of **H1** in Algorithm 1.3.

Algorithm 6.1: Spectral moments $\langle E^1 \rangle$, $\langle E^2 \rangle$, and $\langle E^3 \rangle$ of a one-body Hamiltonian (**moment.py**).

```

1  import numpy as np
2
3  # 1st moment of a one-body Hamiltonian
4  def H1m1(Norb, Ne, h):
5      T1 = np.trace(h)
6      return T1 * Ne/Norb
7
8  # 2nd moment of a one-body Hamiltonian
9  def H1m2(Norb, Ne, h):
10     T1 = np.trace(h)
11     T2 = np.trace(h@h)
12     return (T2*(Norb-Ne) + T1**2*(Ne-1)) * Ne/Norb/(Norb-1)
13
14 # 3rd moment of a one-body Hamiltonian
15 def H1m3(Norb, Ne, h):
16     T1 = np.trace(h)
17     T2 = np.trace(h@h)
18     T3 = np.trace(h@h@h)
19     return (T3 * (Norb-Ne)*(Norb-2*Ne)
20           + T2*T1*3 * (Ne-1)*(Norb-Ne)
21           + T1**3 * (Ne-1)*(Ne-2)) * Ne/Norb/(Norb-1)/(Norb-2)

```

6. Moments of many-electron Hamiltonians

To appreciate the powerfulness (and to confirm the correctness) of the moment formulas, we run a test in Algorithm 6.2. The example problem size is $N_{\text{orb}} = 16$ and $N_e = 8$. We compute $\langle E^1 \rangle$, $\langle E^2 \rangle$, and $\langle E^3 \rangle$ of a one-body Hamiltonian with a randomly generated one-electron symmetric matrix. The script in Algorithm 6.2 computes the moments using both the efficient moment formulas and the expensive many-body approach. A test run gives the following output:

```
Method 1: moment formulas
<E1> = 3.964548
<E2> = 36.860894
<E3> = 313.783669
Time elapsed: 0.000214 seconds

Method 2: full many-body calculation
<E1> = 3.964548
<E2> = 36.860894
<E3> = 313.783669
Time elapsed: 349.701372 seconds
```

Algorithm 6.2: Computing spectral moments $\langle E^1 \rangle$, $\langle E^2 \rangle$, and $\langle E^3 \rangle$ of a one-body Hamiltonian using the efficient moment formulas and the expensive many-electron approach. The file `hamiltonian.py` refers to Algorithm 1.3.

```
1 from moment import H1m1, H1m2, H1m3
2 from hamiltonian import H1
3 import numpy as np
4 import time
5
6 # Problem size
7 (Norb, Ne) = (16, 8)
8
9 # Random symmetric matrix elements
10 def randh(Norb):
11     np.random.seed(0)
12     R = np.random.rand(Norb, Norb)
13     h = np.tril(R) + np.tril(R, -1).T
14     return h
15 h = randh(Norb)
16
17 # Efficient computation using the moment formulas
18 start = time.time()
19 E1 = H1m1(Norb, Ne, h)
20 E2 = H1m2(Norb, Ne, h)
21 E3 = H1m3(Norb, Ne, h)
22 end = time.time()
23 print('Method 1: moment formulas')
24 print('<E1>= %f' % E1)
25 print('<E2>= %f' % E2)
26 print('<E3>= %f' % E3)
27 print('Time elapsed: %f seconds' % (end-start))
28
29 # Expensive computation using the many-electron Hamiltonian
30 start = time.time()
31 H = H1(Norb, Ne, h)
32 E1 = np.trace(H) / len(H)
33 E2 = np.trace(H@H) / len(H)
34 E3 = np.trace(H@H@H) / len(H)
```

```

35 end = time.time()
36 print('Method 2: full many-body calculation')
37 print('<E1>=%f' % E1)
38 print('<E2>=%f' % E2)
39 print('<E3>=%f' % E3)
40 print('Time elapsed: %f seconds' % (end-start))

```

6.2.4. The n -th moment

At this stage, we have described a systematic way to produce $\langle E^n \rangle$ (in terms of the traces of $\mathbf{h}$) for one-body Hamiltonians. In this part, we list the first few moments expressions and summarize the general pattern.

For convenience, we make some simplifications in the notation:

$$\begin{aligned} \text{Tr}^n(\mathbf{h}^m) &\rightarrow [m]^n \\ \begin{pmatrix} N_{\text{orb}} - x \\ N_e - y \end{pmatrix} &\rightarrow \begin{pmatrix} x \\ y \end{pmatrix} \end{aligned}$$

Omitting the prefactor $1/\dim_{\text{H}}$, the first few moments are listed in Table 6.1.

In Table 6.1, the second and the third columns describe a partition problem. For instance, the set $\{A, B, C\}$ can be divided into sub-sets as:

$$\begin{aligned} [3] &\rightarrow \{\{A, B, C\}\} \\ [2][1] &\rightarrow \{\{A, B\}, \{C\}\}, \{\{A, C\}, \{B\}\}, \{\{B, C\}, \{A\}\} \\ [1][1][1] &\rightarrow \{\{A\}, \{B\}, \{C\}\} \end{aligned}$$

Thus $\langle E^3 \rangle$ have the terms: $[3]$, $3 \times [2][1]$, and $[1]^3$. The sequence in the third column follows the “Triangle of multinomial coefficients” (OEIS A080575) [48].

The last column with the combinatorial factors is a little more complicated. For each moment, we collect the first term:

[1]	$\binom{1}{1}$
[2]	$\binom{2}{1}$
[3]	$\binom{3}{1} - \binom{3}{2}$
[4]	$\binom{4}{1} - 4\binom{4}{2} + \binom{4}{3}$
[5]	$\binom{5}{1} - 11\binom{5}{2} + 11\binom{5}{3} - \binom{5}{4}$
[6]	$\binom{6}{1} - 26\binom{6}{2} + 66\binom{6}{3} - 26\binom{6}{4} + \binom{6}{5}$

The table follows the “Triangle of Eulerian numbers” (OEIS A008292) [48] with alternating signs. Further, for product terms like $[5][4]$, the corresponding binomial coefficients

6. Moments of many-electron Hamiltonians

Table 6.1.: The first few moments of a one-body Hamiltonian.

moments	traces	partition factors	combinatorial factors
$\langle E^1 \rangle$	[1]	1	$\binom{1}{1}$
$\langle E^2 \rangle$	[2]	1	$\binom{2}{1}$
	$[1]^2$	1	$\binom{2}{2}$
$\langle E^3 \rangle$	[3]	1	$\binom{3}{1} - \binom{3}{2}$
	$[2][1]$	3	$\binom{3}{3}$
	$[1]^3$	1	$\binom{3}{3}$
$\langle E^4 \rangle$	[4]	1	$\binom{4}{1} - 4\binom{4}{2} + \binom{4}{3}$
	$[3][1]$	4	$\binom{4}{4} - \binom{4}{3}$
	$[2]^2$	3	$\binom{4}{4}$
	$[2][1]^2$	6	$\binom{4}{4}$
	$[1]^4$	1	$\binom{4}{4}$
$\langle E^5 \rangle$	[5]	1	$\binom{5}{1} - 11\binom{5}{2} + 11\binom{5}{3} - \binom{5}{4}$
	$[4][1]$	5	$\binom{5}{5} - 4\binom{5}{3} + \binom{5}{4}$
	$[3][2]$	10	$\binom{5}{5} - \binom{5}{3}$
	$[3][1]^2$	10	$\binom{5}{5} - \binom{5}{4}$
	$[2]^2[1]$	15	$\binom{5}{5}$
	$[2][1]^3$	10	$\binom{5}{5}$
	$[1]^5$	1	$\binom{5}{5}$
			$\binom{5}{5}$
$\langle E^6 \rangle$	[6]	1	$\binom{6}{1} - 26\binom{6}{2} + 66\binom{6}{3} - 26\binom{6}{4} + \binom{6}{5}$
	$[5][1]$	6	$\binom{6}{6} - 11\binom{6}{3} + 11\binom{6}{4} - \binom{6}{5}$
	$[4][2]$	15	$\binom{6}{6} - 4\binom{6}{3} + \binom{6}{4}$
	$[4][1]^2$	15	$\binom{6}{6} - 4\binom{6}{4} + \binom{6}{5}$
	$[3]^2$	10	$\binom{6}{6} - 2\binom{6}{3} + \binom{6}{4}$
	$[3][2][1]$	60	$\binom{6}{6} - \binom{6}{4}$
	$[3][1]^3$	20	$\binom{6}{6} - \binom{6}{5}$
	$[2]^3$	15	$\binom{6}{6}$
	$[2]^2[1]^2$	45	$\binom{6}{6}$
	$[2][1]^4$	15	$\binom{6}{6}$
	$[1]^6$	1	$\binom{6}{6}$

can be constructed in the following way,

$$\begin{array}{|c|c|c|c|}
 \hline
 [5][4] & \binom{4}{1} & -4\binom{4}{2} & \binom{4}{3} \\
 \hline
 -11\binom{5}{1} & -11\binom{9}{2} & 44\binom{9}{3} & -11\binom{9}{4} \\
 11\binom{5}{2} & 11\binom{9}{3} & -44\binom{9}{4} & 11\binom{9}{5} \\
 -\binom{5}{3} & -\binom{9}{4} & 4\binom{9}{5} & -\binom{9}{6}
 \end{array} \Rightarrow \binom{9}{2} - 15\binom{9}{3} + 56\binom{9}{4} - 56\binom{9}{5} + 15\binom{9}{6} - \binom{9}{7}$$

At this stage, we are able to compute any $\langle E^n \rangle$ in terms of the traces of $\mathbf{h}$, for any one-body Hamiltonian.

6.3. Moments of two-body Hamiltonians

In this section, we consider the two-body Hamiltonian, which describes interacting systems:

$$H = \sum_{\substack{\alpha < \beta \\ \gamma < \delta}} \ddot{v}_{\alpha\beta\gamma\delta} c_{\alpha}^{\dagger} c_{\beta}^{\dagger} c_{\gamma} c_{\delta} \quad (6.36)$$

The form of a two-body Hamiltonian is similar to a one-body Hamiltonian squared (6.14). A major difference is in the form of the matrix elements: In (6.14), the matrix elements are products of two one-electron matrix elements; In (6.36), the matrix elements are 4-index tensor elements with additional restrictions. In this chapter, for convenience, we re-write the two-body Hamiltonian in the following form:

$$H = \sum_{aAbB} [aAbB] a^{\dagger} A^{\dagger} b B \quad (6.37)$$

where $(a < A)$ and $(b < B)$ are implied.

Recall (1.22), we can re-order the tensor indices: $[aAbB] = -[aABb]$. On the other hand, we can anti-commute the two annihilators: $bB = -Bb$. Equivalently, we can write (6.37) as:

$$H = \sum_{aABb} [aABb] a^{\dagger} A^{\dagger} B b \quad (6.38)$$

6.3.1. The first moment

First, we work out the first moment:

$$\langle E \rangle = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{aABb} [aABb] \langle i | a^{\dagger} A^{\dagger} B b | i \rangle \quad (6.39)$$

The diagonal density matrix element yields,

$$\langle i | a^{\dagger} A^{\dagger} B b | i \rangle = (\delta_{ab} \delta_{AB} - \delta_{aB} \delta_{Ab}) \Delta(aABb \in i) \quad (6.40)$$

However, the term $\delta_{aB} \delta_{Ab}$ is “self-contradictory”: Because $(a = B)$ implies $(A > B)$ and $(A = b)$ implies $(A < B)$, which are contradictory. Logically,

$$\delta_{aB} \delta_{Ab} = 0 \quad (6.41)$$

The first moment reduces to,

$$\langle E \rangle = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{aABb} [aABb] \delta_{ab} \delta_{AB} \Delta(aABb \in i) = \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{aA \in i} [aAAa] \quad (6.42)$$

With the trick of the conditional combinatorics, we have,

$$\langle E \rangle = \frac{1}{\dim_{\text{H}}} \binom{N_{\text{orb}} - 2}{N_e - 2} \sum_{aA} [aAAa] \quad (6.43)$$

6. Moments of many-electron Hamiltonians

The summation over the paired index can be expressed as the trace of the matrix representation in a two-electron basis. Namely,

$$\sum_{aA} [aAAa] = \text{Tr}(\ddot{\mathbf{v}}) \quad (6.44)$$

For instance, in a 3-orbital-basis system, the matrix representation in a two-electron basis reads,

$$\ddot{\mathbf{v}} = \begin{array}{c|ccc} H & c_1^\dagger c_0^\dagger |0\rangle & c_2^\dagger c_0^\dagger |0\rangle & c_2^\dagger c_1^\dagger |0\rangle \\ \hline \langle 0| c_0 c_1 & [0110] & [0120] & [0121] \\ \langle 0| c_0 c_2 & [0210] & [0220] & [0221] \\ \langle 0| c_1 c_2 & [1210] & [1220] & [1221] \end{array} \quad (6.45)$$

Finally, the first moment of a two-body Hamiltonian:

$$\langle E \rangle = \frac{1}{\dim_{\mathbf{H}}} \binom{N_{\text{orb}}-2}{N_e-2} \text{Tr}(\ddot{\mathbf{v}}) \quad (6.46)$$

The original “**H**-problem” ($\dim_{\mathbf{H}} = N_{\text{orb}}\text{-choose-}N_e$) reduces to a “ **$\ddot{\mathbf{v}}$** -problem” ($\dim_{\ddot{\mathbf{v}}} = N_{\text{orb}}\text{-choose-}2$). If we denote

$$\langle \varepsilon^n \rangle = \frac{1}{\dim_{\ddot{\mathbf{v}}}} \text{Tr}(\ddot{\mathbf{v}}^n) \quad (6.47)$$

Eqn. (6.46) can be equivalently written as,

$$\langle E \rangle = \binom{N_e}{2} \langle \varepsilon \rangle \quad (6.48)$$

The mean eigen-energy of the many-electron system is the mean eigen-energy of the two-electron system times the number of electron-pairs.

6.3.2. The second moment

Next, we consider the Hamiltonian squared:

$$H^2 = \sum_{aABbcCDd} [aABb][cCDd] a^\dagger A^\dagger B b c^\dagger C^\dagger D d \quad (6.49)$$

Re-order the operators,

$$\begin{aligned} a^\dagger A^\dagger B b c^\dagger C^\dagger D d &= a^\dagger A^\dagger B (\delta_{bc} - c^\dagger b) C^\dagger D d \\ &= a^\dagger A^\dagger B C^\dagger D d \delta_{bc} - a^\dagger A^\dagger B c^\dagger b C^\dagger D d \\ &= a^\dagger A^\dagger (\delta_{BC} - C^\dagger B) D d \delta_{bc} - a^\dagger A^\dagger (\delta_{Bc} - c^\dagger B) (\delta_{bc} - C^\dagger b) D d \\ &= a^\dagger A^\dagger D d \delta_{BC} \delta_{bc} - a^\dagger A^\dagger C^\dagger B D d \delta_{bc} \\ &\quad - a^\dagger A^\dagger D d \delta_{Bc} \delta_{bC} + a^\dagger A^\dagger C^\dagger b D d \delta_{Bc} + a^\dagger A^\dagger c^\dagger B D d \delta_{bC} \\ &\quad - a^\dagger A^\dagger c^\dagger b D d \delta_{BC} + a^\dagger A^\dagger c^\dagger C^\dagger B b D d \end{aligned}$$

Note that the term with $\delta_{Bc}\delta_{bC}$ vanishes because of contradiction.

Now, we trace for the second moment:

$$\begin{aligned}
 \langle E^2 \rangle = & \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{aABbcCDd} [aABb][cCDd] (\\
 & \langle i | a^\dagger A^\dagger Dd | i \rangle \delta_{BC} \delta_{bc} \\
 & - \langle i | a^\dagger A^\dagger C^\dagger B Dd | i \rangle \delta_{bc} + \langle i | a^\dagger A^\dagger C^\dagger b Dd | i \rangle \delta_{Bc} \\
 & + \langle i | a^\dagger A^\dagger c^\dagger B Dd | i \rangle \delta_{bC} - \langle i | a^\dagger A^\dagger c^\dagger b Dd | i \rangle \delta_{BC} \\
 & + \langle i | a^\dagger A^\dagger c^\dagger C^\dagger B b Dd | i \rangle \\
 &)
 \end{aligned} \tag{6.50}$$

Expanding each diagonal density matrix element as determinant of deltas, we get,

$$\begin{aligned}
 \langle E^2 \rangle = & \frac{1}{\dim_{\text{H}}} \sum_{i=1}^{\dim_{\text{H}}} \sum_{aABbcCDd} [aABb][cCDd] (\\
 & \begin{vmatrix} \delta_{ad} & \delta_{aD} \\ \delta_{Ad} & \delta_{AD} \end{vmatrix} \Delta(aADd \in i) \delta_{BC} \delta_{bc} \\
 & - \begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{aB} \\ \delta_{Ad} & \delta_{AD} & \delta_{AB} \\ \delta_{Cd} & \delta_{CD} & \delta_{CB} \end{vmatrix} \Delta(aACB Dd \in i) \delta_{bc} + \begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{ab} \\ \delta_{Ad} & \delta_{AD} & \delta_{Ab} \\ \delta_{Cd} & \delta_{CD} & \delta_{Cb} \end{vmatrix} \Delta(aACb Dd \in i) \delta_{Bc} \\
 & + \begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{aB} \\ \delta_{Ad} & \delta_{AD} & \delta_{AB} \\ \delta_{cd} & \delta_{cD} & \delta_{cB} \end{vmatrix} \Delta(aAcB Dd \in i) \delta_{bC} - \begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{ab} \\ \delta_{Ad} & \delta_{AD} & \delta_{Ab} \\ \delta_{cd} & \delta_{cD} & \delta_{cb} \end{vmatrix} \Delta(aAc b Dd \in i) \delta_{BC} \\
 & + \begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{ab} & \delta_{aB} \\ \delta_{Ad} & \delta_{AD} & \delta_{Ab} & \delta_{AB} \\ \delta_{cd} & \delta_{cD} & \delta_{cb} & \delta_{cB} \\ \delta_{Cd} & \delta_{CD} & \delta_{Cb} & \delta_{CB} \end{vmatrix} \Delta(aAcCBb Dd \in i) \\
 &)
 \end{aligned} \tag{6.51}$$

Now, term-by-term! (in the following results, all the contradictory terms are dropped)

The first term:

$$\begin{vmatrix} \delta_{ad} & \delta_{aD} \\ \delta_{Ad} & \delta_{AD} \end{vmatrix} \delta_{BC} \delta_{bc} = (\delta_{ad} \delta_{AD}) \delta_{BC} \delta_{bc}$$

6. Moments of many-electron Hamiltonians

The four 3-by-3 determinants:

$$\begin{aligned}
\begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{aB} \\ \delta_{Ad} & \delta_{AD} & \delta_{AB} \\ \delta_{Cd} & \delta_{CD} & \delta_{CB} \end{vmatrix} \delta_{bc} &= (\delta_{ad}\delta_{AD}\delta_{CB} + \delta_{aD}\delta_{AB}\delta_{Cd} + \delta_{aB}\delta_{Ad}\delta_{CD} - \delta_{aB}\delta_{AD}\delta_{Cd} - \delta_{ad}\delta_{AB}\delta_{CD})\delta_{bc} \\
\begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{ab} \\ \delta_{Ad} & \delta_{AD} & \delta_{Ab} \\ \delta_{Cd} & \delta_{CD} & \delta_{Cb} \end{vmatrix} \delta_{Bc} &= (\delta_{ab}\delta_{Ad}\delta_{CD} - \delta_{ab}\delta_{AD}\delta_{Cd} - \delta_{ad}\delta_{Ab}\delta_{CD})\delta_{Bc} \\
\begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{aB} \\ \delta_{Ad} & \delta_{AD} & \delta_{AB} \\ \delta_{cd} & \delta_{cD} & \delta_{cB} \end{vmatrix} \delta_{bC} &= (\delta_{ad}\delta_{AB}\delta_{cd} - \delta_{aB}\delta_{AD}\delta_{cd} - \delta_{ad}\delta_{AB}\delta_{cD})\delta_{bC} \\
\begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{ab} \\ \delta_{Ad} & \delta_{AD} & \delta_{Ab} \\ \delta_{cd} & \delta_{cD} & \delta_{cb} \end{vmatrix} \delta_{BC} &= (\delta_{ad}\delta_{AD}\delta_{cb} + \delta_{aD}\delta_{Ab}\delta_{cd} + \delta_{ab}\delta_{Ad}\delta_{cD} - \delta_{ab}\delta_{AD}\delta_{cd} - \delta_{ad}\delta_{Ab}\delta_{cD})\delta_{BC}
\end{aligned}$$

The 4-by-4 determinant term:

$$\begin{aligned}
&\begin{vmatrix} \delta_{ad} & \delta_{aD} & \delta_{ab} & \delta_{aB} \\ \delta_{Ad} & \delta_{AD} & \delta_{Ab} & \delta_{AB} \\ \delta_{cd} & \delta_{cD} & \delta_{cb} & \delta_{cB} \\ \delta_{Cd} & \delta_{CD} & \delta_{Cb} & \delta_{CB} \end{vmatrix} \\
&= \delta_{ad} \begin{vmatrix} \delta_{AD} & \delta_{Ab} & \delta_{AB} \\ \delta_{cD} & \delta_{cb} & \delta_{cB} \\ \delta_{CD} & \delta_{Cb} & \delta_{CB} \end{vmatrix} - \delta_{aD} \begin{vmatrix} \delta_{Ad} & \delta_{Ab} & \delta_{AB} \\ \delta_{cd} & \delta_{cb} & \delta_{cB} \\ \delta_{Cd} & \delta_{Cb} & \delta_{CB} \end{vmatrix} + \delta_{ab} \begin{vmatrix} \delta_{Ad} & \delta_{AD} & \delta_{AB} \\ \delta_{cd} & \delta_{cD} & \delta_{cB} \\ \delta_{Cd} & \delta_{CD} & \delta_{CB} \end{vmatrix} - \delta_{aB} \begin{vmatrix} \delta_{Ad} & \delta_{AD} & \delta_{Ab} \\ \delta_{cd} & \delta_{cD} & \delta_{cb} \\ \delta_{Cd} & \delta_{CD} & \delta_{Cb} \end{vmatrix} \\
&= \delta_{ad}(\delta_{AD}\delta_{cb}\delta_{CB} + \delta_{Ab}\delta_{cB}\delta_{CD} + \delta_{AB}\delta_{cD}\delta_{Cb} - \delta_{AB}\delta_{cb}\delta_{CD} - \delta_{Ab}\delta_{cD}\delta_{CB}) \\
&\quad - \delta_{aD}(\delta_{AB}\delta_{cd}\delta_{Cb} - \delta_{AB}\delta_{cb}\delta_{Cd} - \delta_{Ab}\delta_{cd}\delta_{CB}) \\
&\quad + \delta_{ab}(\delta_{Ad}\delta_{cD}\delta_{CB} + \delta_{AD}\delta_{cB}\delta_{Cd} + \delta_{AB}\delta_{cd}\delta_{CD} - \delta_{AD}\delta_{cd}\delta_{CB} - \delta_{Ad}\delta_{cB}\delta_{CD}) \\
&\quad - \delta_{aB}(\delta_{AD}\delta_{cb}\delta_{Cd} - \delta_{AD}\delta_{cd}\delta_{Cb} - \delta_{Ad}\delta_{cb}\delta_{CD})
\end{aligned}$$

One must pay attention when combining the elements' indices (which have implicit constraints $a < A$, $b < B$, $c < C$, $d < D$) with the Kronecker deltas. For example,

$$\sum_{aD} [aABb][cCDd]\delta_{aD} = \sum_a [aABb][cCad] \quad \text{with the constraint } (a > d)$$

Originally, the indices a and d are independent. But, in the example above, it gets an additional constraint because a replaces D and $D > d$. Such additional constraints are crucial for merging the terms in the next step. Now, for convenience, we write,

$$\sum_{aD} [aABb][cCDd]\delta_{aD} = \sum_a [aABb][cCa > d]$$

Now, we continue Eqn. (6.51) by combining the Kronecker deltas. Carefully collecting

the terms and using the symmetries (1.22), we obtain,

$$\begin{aligned}
\langle E^2 \rangle = & \frac{1}{\dim_{\mathcal{H}}} \sum_{i=1}^{\dim_{\mathcal{H}}} (\\
& \sum_{aA \in i} \sum_{Bb} [aABb][bBAa] - \sum_{aAB \in i} \sum_b [aABb][bBAa] - \sum_{aAb \in i} \sum_B [aABb][bBAa] + \sum_{aABb \in i} [aABb][bBAa] \\
& + \sum_{aAC \in i} \sum_c (\\
& \underbrace{[aAA > c][cCC > a] + [aAA > c][cCC < a] + [aAA < c][cCC > a]}_{=[aAAc][cCCA] \quad \text{note: } [aAA < c][cCC < a]=0 \text{ because of contradiction}} \\
& + \underbrace{[Aaa > c][cCC > A] + [Aaa > c][cCC < A] + [Aaa < c][cCC > A] + [Aaa < c][cCC < A]}_{=[Aaac][cCCA]} \\
&) \\
& + \sum_{aAc \in i} \sum_C (\\
& \underbrace{[aAA > C][Ccc > a] + [aAA > C][Ccc < a] + [aAA < C][Ccc > a] + [aAA < C][Ccc < a]}_{=[aAAC][Ccca]} \\
& + \underbrace{[Aaa > C][Ccc < A] + [Aaa < C][Ccc > A] + [Aaa < C][Ccc < A]}_{=[AaaC][CccA] \quad \text{note: } [Aaa > C][Ccc > A]=0 \text{ because of contradiction}} \\
&) \\
& + \sum_{aAcC \in i} (\\
& \underbrace{[aAA > c][cCC > a] + [aAA > c][cCC < a] + [aAA < c][cCC > a]}_{=[aAAc][cCCA]} \\
& + \underbrace{[Aaa > c][cCC > A] + [Aaa > c][cCC < A] + [Aaa < c][cCC > A] + [Aaa < c][cCC < A]}_{=[Aaac][cCCA]} \\
& + \underbrace{[aAA > C][Ccc > a] + [aAA > C][Ccc < a] + [aAA < C][Ccc > a] + [aAA < C][Ccc < a]}_{=[aAAC][Ccca]} \\
& + \underbrace{[Aaa > C][Ccc < A] + [Aaa < C][Ccc > A] + [Aaa < C][Ccc < A]}_{=[AaaC][CccA]} \\
&) \\
& + \sum_{aAcC \in i} [aAAa][cCCc] \\
&)
\end{aligned} \tag{6.52}$$

6. Moments of many-electron Hamiltonians

Using the trick of the conditional combinatorics, we have,

$$\begin{aligned}
& \langle E^2 \rangle \\
&= \frac{1}{\dim_{\text{H}}} \left(\right. \\
& \quad \left[\binom{N_{\text{orb}}-2}{N_e-2} - 2 \binom{N_{\text{orb}}-3}{N_e-3} + \binom{N_{\text{orb}}-4}{N_e-4} \right] \sum_{aABb} [aABb][bBAa] \\
& \quad + \left[\binom{N_{\text{orb}}-3}{N_e-3} - \binom{N_{\text{orb}}-4}{N_e-4} \right] \sum_{aAcC} [aAAc][cCCa] + [aAAC][Ccca] + [Aaac][cCCA] + [AaaC][CccA] \\
& \quad + \binom{N_{\text{orb}}-4}{N_e-4} \sum_{aAcC} [aAAa][cCCc] \\
& \left. \right) \\
&= \frac{1}{\dim_{\text{H}}} \left(\right. \\
& \quad \binom{N_{\text{orb}}-4}{N_e-2} \sum_{aABb} [aABb][bBAa] \\
& \quad + \binom{N_{\text{orb}}-4}{N_e-3} \sum_{aAcC} [aAAc][cCCa] + [aAAC][Ccca] + [Aaac][cCCA] + [AaaC][CccA] \\
& \quad + \binom{N_{\text{orb}}-4}{N_e-4} \sum_{aAcC} [aAAa][cCCc] \\
& \left. \right) \tag{6.53}
\end{aligned}$$

In a compact notation, we write,

$$\boxed{\langle E^2 \rangle = \frac{1}{\dim_{\text{H}}} \left[\binom{N_{\text{orb}}-4}{N_e-2} \text{Tr}(\ddot{\mathbf{v}}^2) + \binom{N_{\text{orb}}-4}{N_e-3} X + \binom{N_{\text{orb}}-4}{N_e-4} \text{Tr}^2(\ddot{\mathbf{v}}) \right]} \tag{6.54}$$

where,

$$X = \sum_{\substack{\alpha < \beta \\ \gamma < \delta}} \ddot{v}_{\alpha\beta\beta\gamma} \ddot{v}_{\gamma\delta\delta\alpha} + \ddot{v}_{\alpha\beta\beta\delta} \ddot{v}_{\delta\gamma\gamma\alpha} + \ddot{v}_{\beta\alpha\alpha\gamma} \ddot{v}_{\gamma\delta\delta\beta} + \ddot{v}_{\beta\alpha\alpha\delta} \ddot{v}_{\delta\gamma\gamma\beta} \tag{6.55}$$

In fact, if we play a bit with the indices, the X -term can be re-written as,

$$X = \left(\sum_{\substack{\alpha < \beta \\ \gamma < \delta}} + \sum_{\substack{\alpha < \beta \\ \gamma > \delta}} + \sum_{\substack{\alpha > \beta \\ \gamma < \delta}} + \sum_{\substack{\alpha > \beta \\ \gamma > \delta}} \right) \ddot{v}_{\alpha\beta\beta\gamma} \ddot{v}_{\gamma\delta\delta\alpha} = \sum_{\alpha\beta\gamma\delta} \ddot{v}_{\alpha\beta\beta\gamma} \ddot{v}_{\gamma\delta\delta\alpha} \tag{6.56}$$

The matrix elements have a special form that the two middle indices are equal. The physical picture is that during the pair interactions, one of the electron state remains untouched. This is in some sense a one-body interaction. Indeed, since the two middle indices are summed over independently, we introduce,

$$\ddot{h}_{\alpha\beta} \equiv \sum_i \ddot{v}_{\alpha ii\beta} \tag{6.57}$$

Thus, the X -term reads,

$$X = \text{Tr}(\ddot{\mathbf{h}}^2) \quad (6.58)$$

Algorithm 6.3 gives example codes for computing the spectral moments $\langle E^1 \rangle$ and $\langle E^2 \rangle$ of a two-body Hamiltonian for a many-electron system. Note that the input arguments of the functions are identical to the arguments of H2 in Algorithm 1.3.

Algorithm 6.3: Spectral moments $\langle E^1 \rangle$ and $\langle E^2 \rangle$ of a two-body Hamiltonian (`moment.py`).

```

1  import numpy as np
2
3  # 1st moment of a two-body Hamiltonian
4  def H2m1(Norb, Ne, vee):
5      T1 = 0
6      idx = ((a,b) for a in range(Norb) for b in range(Norb) if a<b)
7      for (a,b) in idx:
8          T1 += vee[a,b,b,a]
9      return T1 * Ne*(Ne-1)/Norb/(Norb-1)
10
11 # 2nd moment of a two-body Hamiltonian
12 def H2m2(Norb, Ne, vee):
13     T1 = T2 = 0
14     idx = [(a,b) for a in range(Norb) for b in range(Norb) if a<b]
15     for (a,b) in idx:
16         T1 += vee[a,b,b,a]
17         for (c,d) in idx:
18             T2 += vee[a,b,d,c]*vee[c,d,b,a]
19
20     hee = np.trace(vee, axis1=1, axis2=2) # hee[a,b] = np.trace(vee[a,:,:,:b])
21     X = np.trace(hee@hee)
22
23     return (T2 * (Norb-Ne)*(Norb-Ne-1)
24           + X * (Ne-2)*(Norb-Ne)
25           + T1**2 * (Ne-2)*(Ne-3)) * Ne*(Ne-1)/Norb/(Norb-1)/(Norb-2)/(Norb-3)

```

To appreciate the powerfulness (and to confirm the correctness) of the moment formulas, we run a test in Algorithm 6.4. The example problem size is $N_{\text{orb}} = 16$ and $N_e = 8$. We compute $\langle E^1 \rangle$ and $\langle E^2 \rangle$ of a two-body Hamiltonian with a randomly generated two-electron symmetric tensor. The script in Algorithm 6.4 computes the moments using both the efficient moment formulas and the expensive many-body approach. A test run gives the following output:

```

Method 1: moment formulas
<E1> = -13.979175
<E2> = 841.826378
Time elapsed: 0.009434 seconds

```

```

Method 2: full many-body calculation
<E1> = -13.979175
<E2> = 841.826378
Time elapsed: 456.797280 seconds

```

6. Moments of many-electron Hamiltonians

Algorithm 6.4: Computing spectral moments $\langle E^1 \rangle$ and $\langle E^2 \rangle$ of a two-body Hamiltonian using the efficient moment formulas and the expensive many-electron approach. The file `hamiltonian.py` refers to Algorithm 1.3.

```

1 from moment import H2m1, H2m2
2 from hamiltonian import H2
3 import numpy as np
4 import time
5
6 # Problem size
7 (Norb, Ne) = (16, 8)
8
9 # Random symmetric tensor elements
10 def randvee(Norb):
11     np.random.seed(0)
12     vee = np.zeros((Norb, Norb, Norb, Norb))
13     idx = ((a, b, c, d) for a in range(Norb) for b in range(Norb) if a < b
14             for c in range(Norb) for d in range(Norb) if c < d)
15     for (a, b, c, d) in idx:
16         r = np.random.rand()
17         vee[a, b, c, d] = vee[b, a, d, c] = vee[c, d, a, b] = vee[d, c, b, a] = r
18         vee[a, b, d, c] = vee[b, a, c, d] = vee[c, d, b, a] = vee[d, c, a, b] = -r
19     return vee
20 vee = randvee(Norb)
21
22 # Efficient computation using the moment formulas
23 start = time.time()
24 E1 = H2m1(Norb, Ne, vee)
25 E2 = H2m2(Norb, Ne, vee)
26 end = time.time()
27 print('Method 1: moment formulas')
28 print('<E1> = %f' % E1)
29 print('<E2> = %f' % E2)
30 print('Time elapsed: %f seconds' % (end - start))
31
32 # Expensive computation using the many-electron Hamiltonian
33 start = time.time()
34 H = H2(Norb, Ne, vee)
35 E1 = np.trace(H) / len(H)
36 E2 = np.trace(H@H) / len(H)
37 end = time.time()
38 print('Method 2: full many-body calculation')
39 print('<E1> = %f' % E1)
40 print('<E2> = %f' % E2)
41 print('Time elapsed: %f seconds' % (end - start))

```

Eqn. (6.54) is the 2nd-moment formula for a *general* two-body Hamiltonian. A typical example is the Coulomb Hamiltonian for a given system. In Chapter 5, we computed the variances of Coulomb Hamiltonians for atomic open-shell systems (from a many-body approach). From Eqns. (6.46) and (6.54) we can now derive the variances of different open-shell systems in terms of the Slater-Condon parameters and the Gaunt coefficients explicitly (see Table B.5 in Appendix B).

At this stage, we are able to – for instance – compare the interaction strengths (by comparing the variances) between the spin-orbit (one-body) and the Coulomb (two-body) interactions for a given system *without* doing a many-body calculation. This is a big milestone of attacking the “impossible” many-body problems. In this chapter, we managed to solve a general one-body Hamiltonian up to the n -th moment, where n can be any

6.3. Moments of two-body Hamiltonians

non-negative integer¹; we also managed to solve a general two-body Hamiltonian up to the 2nd moment. Higher-order moments should be possible to derive although it would require a lot of patience. Moreover, moments of general Hamiltonians consisting *both* one-body *and* two-body terms can be further investigated.

¹The zeroth moment, which we didn't mention, is trivially $\langle E^0 \rangle = \frac{1}{\dim_{\mathcal{H}}} \sum_{i=1}^{\dim_{\mathcal{H}}} \langle i | i \rangle = 1$.

7. Recursive computation of Gaunt coefficients

7.1. Gaunt coefficients

Gaunt coefficients are purely solid angle integrals over three spherical harmonics [49]:

$$\langle Y_{l_1 m_1} | Y_{k \mu} | Y_{l_2 m_2} \rangle = \int_0^{2\pi} d\phi \int_0^\pi d\theta \sin \theta \overline{Y_{l_1 m_1}}(\theta, \phi) Y_{k \mu}(\theta, \phi) Y_{l_2 m_2}(\theta, \phi) \quad (7.1)$$

Here we use a simplified notation:

$$\langle Y_{l_1 m_1} | Y_{k \mu} | Y_{l_2 m_2} \rangle \mapsto \langle l_1 m_1 | k \mu | l_2 m_2 \rangle \quad (7.2)$$

Gaunt coefficients are ubiquitous for problems that involve spherical harmonic expansions [50]. Previously we have seen Gaunt coefficients in the Coulomb matrix elements (4.12). In Chapter 8, we will encounter Gaunt coefficients for evaluating orbital overlaps. In Chapter 9, we will need the coefficients till large angular quantum numbers for the so called re-centering technique.

The purpose of this chapter is to develop a systematic approach based on finite-precision arithmetic to compute the Gaunt coefficients efficiently and accurately, even for large angular momenta.

7.2. Symmetry properties

Before attempting to compute a Gaunt coefficient, the first question to ask is: “Is this Gaunt coefficient trivial?” For a given Gaunt coefficient:

$$\langle l_1 m_1 | k \mu | l_2 m_2 \rangle$$

It is non-trivial¹ if the following conditions (the selection rules) are satisfied:

$$\mu = m_1 - m_2 \quad (7.3)$$

$$|l_1 - l_2| \leq k \leq l_1 + l_2 \quad \text{and} \quad l_1 + l_2 + k \text{ is even} \quad (7.4)$$

¹It would be less appropriate to call it non-zero, since a non-trivial Gaunt coefficient may happen to have a zero value.

7. Recursive computation of Gaunt coefficients

If the selection rules are not fulfilled, the Gaunt coefficients are simply zeros. This property greatly reduces our computational effort. Notice that the condition (7.3) restricts μ to take one unique value. Thus it is normally sufficient to specify m_1 and m_2 .

There are several symmetry properties of Gaunt coefficients which can also simplify the computation, for instance,

- Gaunt coefficients are real numbers

$$\langle l_1 m_1 | k \mu | l_2 m_2 \rangle = \overline{\langle l_1 m_1 | k \mu | l_2 m_2 \rangle} \quad (7.5)$$

- Inverse symmetry

$$\langle l_1 m_1 | k \mu | l_2 m_2 \rangle = \langle l_1, -m_1 | k, -\mu | l_2, -m_2 \rangle \quad (7.6)$$

- Transpose symmetry ($\overline{k\mu}$ denotes $\overline{Y_{k\mu}}$)

$$\langle l_1 m_1 | k \mu | l_2 m_2 \rangle = \langle l_2 m_2 | \overline{k\mu} | l_1 m_1 \rangle \quad (7.7)$$

7.3. Explicit formula

There is a close relation between the Wigner $3j$ symbols and the integral of three spherical harmonics (note that the $Y_{l_1 m_1}$ below is without complex conjugate) [51]:

$$\begin{aligned} & \int_0^{2\pi} d\phi \int_0^\pi d\theta \sin \theta Y_{l_1 m_1}(\theta, \phi) Y_{l_2 m_2}(\theta, \phi) Y_{l_3 m_3}(\theta, \phi) \\ &= \sqrt{\frac{(2l_1+1)(2l_2+1)(2l_3+1)}{4\pi}} \begin{pmatrix} l_1 & l_2 & l_3 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \end{aligned} \quad (7.8)$$

Eqn. (7.8) can be further expanded more explicitly [52, 53]:

$$\begin{aligned} & \int_0^{2\pi} d\phi \int_0^\pi d\theta \sin \theta Y_{l_1 m_1}(\theta, \phi) Y_{l_2 m_2}(\theta, \phi) Y_{l_3 m_3}(\theta, \phi) \\ &= \Delta(m_1+m_2+m_3=0) \Delta(|l_1-l_2| \leq l_3 \leq l_1+l_2) \Delta(l_1+l_2+l_3 \text{ is even}) \\ & \times (-1)^{L+l_3+m_1-m_2} \sqrt{\frac{(2l_1+1)(2l_2+1)(2l_3+1)}{4\pi}} \\ & \times \frac{L!}{(2L+1)!} \frac{(-l_1+l_2+l_3)!(l_1-l_2+l_3)!(l_1+l_2-l_3)!}{(L-l_1)!(L-l_2)!(L-l_3)!} \\ & \times \sqrt{(l_1-m_1)!(l_1+m_1)!(l_2-m_2)!(l_2+m_2)!(l_3-m_3)!(l_3+m_3)!} \\ & \times \sum_{\gamma=\gamma_{\min}}^{\gamma_{\max}} \frac{(-1)^\gamma}{\gamma!(l_3-l_1-m_2+\gamma)!(l_3-l_2+m_1+\gamma)!(l_1-m_1-\gamma)!(l_2+m_2-\gamma)!(l_1+l_2-l_3-\gamma)!} \end{aligned} \quad (7.9)$$

where $L = \frac{1}{2}(l_1 + l_2 + l_3)$ and $\Delta(\text{condition})$ returns 1 if the condition is true, otherwise 0. The index γ runs over all integer values where no negative factorial number occurs. Equivalently,

$$\gamma_{\min} = \max(0, -l_3 + l_1 + m_2, -l_3 + l_2 - m_1)$$

$$\gamma_{\max} = \min(l_1 - m_1, l_2 + m_2, l_1 + l_2 - l_3)$$

The explicit formula (7.9) is in fact unsuitable for finite-precision arithmetic, because the sum over γ involves large factorials with alternating signs. Such operations can cause the loss of significance in the floating-point arithmetic.

Handling large factorials with finite-precision arithmetic:

Suppose we want to compute, say,

$$\frac{25! \times 50!}{32! \times \sqrt{69!}}$$

To avoid numerical overflow or underflow arising from the factorials in an intermediate stage, we transform the problem using natural logarithm. Without individually evaluating each factorial, we first combine the terms under the logarithm scale. After the combination, we take exponential to bring out the final result. Equivalently, we evaluate,

$$\exp\left(\ln 25! + \ln 50! - \ln 32! - \frac{1}{2} \ln 69!\right)$$

Example Python code:

```
1 import math
2 import numpy as np
3
4 logfac = np.append(0, np.cumsum(np.log(np.arange(1, 70)))) # [ln(0!) ln(1!) ln(2!) ...]
5 result = math.exp(logfac[25] + logfac[50] - logfac[32] - logfac[69]/2)
```

An implementation of the explicit formula is given in Algorithm 7.1.

Algorithm 7.1: Compute Gaunt coefficients using the explicit formula.

```
1 import math
2 import numpy as np
3
4 # Integrate Y_{l1m1} Y_{l2m2} Y_{l3m3}
5 def I3Y(l1, l2, l3, m1, m2, m3):
6     # Trivial cases
7     if l1 < abs(m1) or l2 < abs(m2) or l3 < abs(m3) or (m1+m2+m3) != 0:
8         return 0
9     if not abs(l1-l2) <= l3 <= (l1+l2) or (l1+l2+l3)%2 != 0:
10        return 0
11
12    # Logarithmic factorial
13    logfac = np.append(0, np.cumsum(np.log(np.arange(1, 2*(l1+l2)+2))))
14
15    # Factorials in front of the sum
16    L = (l1+l2+l3)//2
17    gfac = logfac[L]+logfac[-l1+l2+l3]+logfac[l1-l2+l3]+logfac[l1+l2-l3] \
```

7. Recursive computation of Gaunt coefficients

```

18     - logfac[2*L+1]-logfac[L-11]-logfac[L-12]-logfac[L-13]          \
19     + (logfac[l1-m1]+logfac[l2-m2]+logfac[l3-m3])                  \
20     + logfac[l1+m1]+logfac[l2+m2]+logfac[l3+m3])/2
21
22     # Calculate the sum
23     gsum = 0
24     imin = max(0, l1-l3+m2, l2-l3-m1)
25     imax = min(l2+m2, l1-m1, l1+l2-l3)
26     for i in range(imin, imax+1):
27         gsum += (-1)**i * math.exp(gfac
28             - logfac[l3-l1-m2+i]-logfac[l1-m1-i]-logfac[i]          \
29             - logfac[l3-l2+m1+i]-logfac[l2+m2-i]-logfac[l1+l2-l3-i]) \
30
31     return (-1)**(L+l3+m1-m2)*math.sqrt((2*l1+1)*(2*l2+1)*(2*l3+1)/(4*math.pi))*gsum
32
33 # Integrate conjugate(Y_{l1m1}) Y_{k,m1-m2} Y_{l2m2}
34 def Gaunt(l1, l2, k, m1, m2):
35     return (-1)**m1*I3Y(l1, k, l2, -m1, m1-m2, m2)

```

To test the accuracy of the numerical results produced by the explicit algorithm, we implemented a symbolic version in Python using the `sympy` library as our reference.

For a given $l_{\max}$, we compute *all* non-trivial Gaunt coefficients with $l_1 = l_{\max}$, $l_2 \leq l_1$, and for all possible k 's (according to the selection rule (7.4)). For instance, if $l_{\max} = 2$, we compute Gaunt coefficients:

$$\begin{array}{ccc}
 \langle 2m_1 | 2\mu | 0m_2 \rangle & \langle 2m_1 | 1\mu | 1m_2 \rangle & \langle 2m_1 | 3\mu | 1m_2 \rangle \\
 \langle 2m_1 | 0\mu | 2m_2 \rangle & \langle 2m_1 | 2\mu | 2m_2 \rangle & \langle 2m_1 | 4\mu | 2m_2 \rangle
 \end{array}$$

with all possible (m_1, m_2) and $\mu = m_1 - m_2$, which contain in total 78 non-trivial Gaunt coefficients.

A typical Gaunt coefficient has a value between -1 to 1 . Since there are non-trivial Gaunt coefficients with zero or near-zero values, we compute the absolute errors of the Gaunt matrices. For a each $l_{\max}$, we ask for the maximum absolute error among all Gaunt coefficients calculated. We tested the explicit algorithm for $l_{\max}$ from 0 to 65. The error plot is shown in Fig. 7.1. The error grows exponentially with $l_{\max}$ (linear in log scale). At $l_{\max} = 65$, the explicit implementation is accurate to about 2 or 3 decimal points.

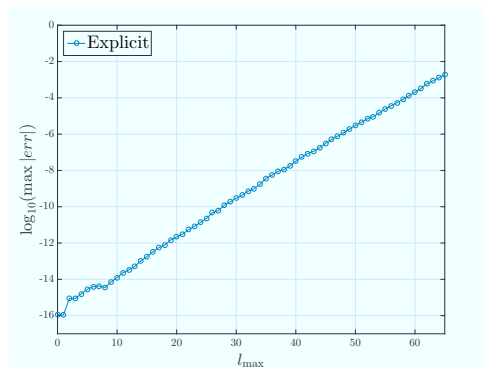


Figure 7.1.: Maximum absolute errors (in $\log_{10}$ scale) of all Gaunt coefficients computed with the explicit formula using double-precision floating-point format.

7.4. Gaunt coefficients arranged as a matrix

Given l_1 , l_2 , and k , we can arrange the Gaunt coefficients in a matrix form [15]. As an illustration, we consider the Gaunt coefficient matrix for $l_1 = 1$ and $l_2 = 2$ (for each k there is one such matrix). This matrix has the following rectangular shape:

-1					
0					
1					
	-2	-1	0	1	2

where the m_1 index traverses vertically (row number) and the m_2 index traverses horizontally (column number). Notice that μ is determined uniquely by $m_1 - m_2$ for non-trivial elements.

To get an intuitive understanding of the Gaunt matrices, Fig. 7.2 visualizes three examples with $l_1 = l_2 = 10$ and $k = 4, 10, 14$, respectively. The values of each element are mapped to red-blue colors, where red ones indicate positive values and blue ones indicate negative values. Notice that the elements with $k < |\mu|$ are zeros: if $k = 0$, the matrix is simply diagonal; as k increases, the non-trivial region becomes broader. It is important to realize that the Gaunt coefficients are highly oscillatory. This becomes a major problem when we evaluate them numerically with finite-precision arithmetic.

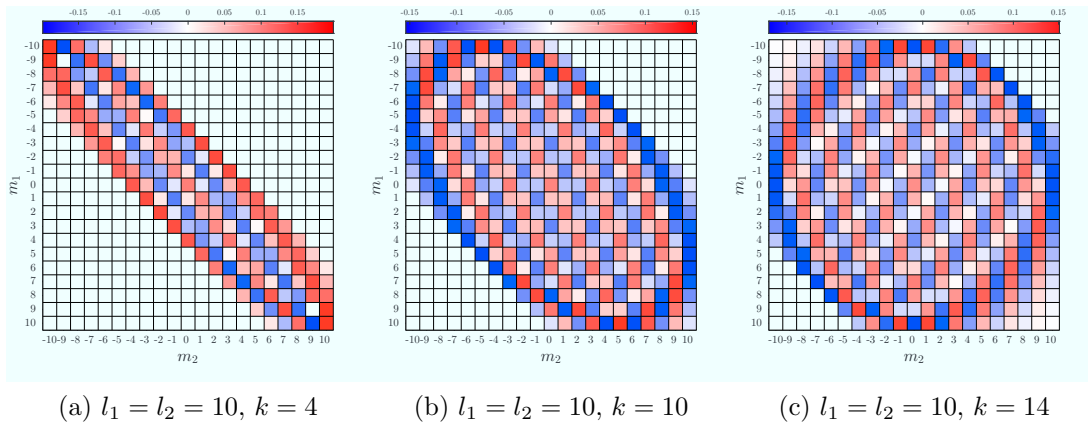


Figure 7.2.: Typical structures of Gaunt matrices. (a) Gaunt matrix with $l_1 = l_2 = 10$ and $k = 4$; (b) Gaunt matrix with $l_1 = l_2 = 10$ and $k = 10$; (c) Gaunt matrix with $l_1 = l_2 = 10$ and $k = 14$; Red colors indicate positive values and blue colors indicate negative values. The matrix elements are highly oscillatory. Elements with $k < |\mu|$ are zeros: the “non-trivial band” expands as k increases.

To fill up a matrix, naively, we would compute $(2l_1 + 1)(2l_2 + 1)$ number of elements. But the inverse symmetry (7.6) helps to reduce half of the computation. While the central

7. Recursive computation of Gaunt coefficients

element $(m_1=0, m_2=0)$ is unique, all the others are in pairs: $(m_1, m_2) = (-m_1, -m_2)$. Moreover, we do not need to compute elements with $k < |\mu|$, as they are simply zeros. A Python routine for filling up a Gaunt matrix is given in Algorithm 7.2.

Algorithm 7.2: Filling up a Gaunt matrix.

```

1  # Create a Gaunt matrix
2  # (-l1,-l2) ... (-l1, l2)
3  #      ...      ...      ...
4  # ( l1,-l2) ... ( l1, l2)
5  def Gaunt_mat(l1, l2, k):
6      G = np.zeros((2*l1+1, 2*l2+1))
7
8      # Diagonal mu=0
9      for m in range(min(l1,l2)+1):
10         G[l1-m, l2-m] = G[l1+m, l2+m] = Gaunt(l1, l2, k, m, m)
11
12     # Off-diagonal mu=1,2,...,k
13     for mu in range(1, k+1):
14         for m1 in range(max(-l2+mu,-l1), min(l2+mu,l1)+1):
15             m2 = m1-mu
16             G[l1-m1, l2-m2] = G[l1+m1, l2+m2] = Gaunt(l1, l2, k, m1, m2)
17     return G

```

7.5. Recursion relations

Gaunt coefficients consist of three spherical harmonics which can be expressed in associated Legendre polynomials. From the existing recursion relations of associated Legendre polynomials, one can derive the corresponding recursion relations for Gaunt coefficients. The detailed derivations are shown in the end of this section. Here we put the recursions directly:

$$\langle l_1 m_1 | k \mu | l_2 m_2 \rangle = A \langle l_1 - 1, m_1 | k - 1, \mu | l_2 m_2 \rangle + B \langle l_1 + 1, m_1 | k - 1, \mu | l_2 m_2 \rangle + C \langle l_1 m_1 | k - 2, \mu | l_2 m_2 \rangle \quad (7.10)$$

where,

$$A = \sqrt{\frac{(2k+1)(2k-1)(l_1+m_1)(l_1-m_1)}{(k+\mu)(k-\mu)(2l_1+1)(2l_1-1)}}, \quad B = \sqrt{\frac{(2k+1)(2k-1)(l_1+m_1+1)(l_1-m_1+1)}{(k+\mu)(k-\mu)(2l_1+3)(2l_1+1)}}, \quad C = -\sqrt{\frac{(2k+1)(k+\mu-1)(k-\mu-1)}{(k+\mu)(k-\mu)(2k-3)}}$$

Normally, a recursive function implies exponential computational effort. Because each node branches into several nodes, as the iteration goes deeper, the “tree” grows exponentially. Here, the branching nodes are aligned on a grid with l_1 and k as coordinates (notice that the recursive indices are l_1 and k only). This recursion stops when $k = |\mu|$. If $k = \mu = 0$, $\langle l_1 m_1 | 00 | l_2 m_2 \rangle$ is simply $\frac{1}{\sqrt{4\pi}} \delta_{l_1 l_2} \delta_{m_1 m_2}$. But, in general, we have to solve $\langle l_1 m_1 | k, \pm k | l_2 m_2 \rangle$ such that we can have starting points for the recursion. Pictorially, we illustrate the recursion scheme for $\langle 32 | 81 | 51 \rangle$ in Fig. 7.3. All empty circles indicate zero elements. Only the solid circles are the non-trivial Gaunt coefficients to compute. The blue dashed lines enclose the region where the selection rules are fulfilled. The red dashed lines mark the boundaries where $l = |m|$ for Y_{lm} . To reach the target $\langle 32 | 81 | 51 \rangle$, we start from the base cases $\langle 42 | 11 | 51 \rangle$ and $\langle 62 | 11 | 51 \rangle$, then travel along the grid towards $\langle 32 | 81 | 51 \rangle$.

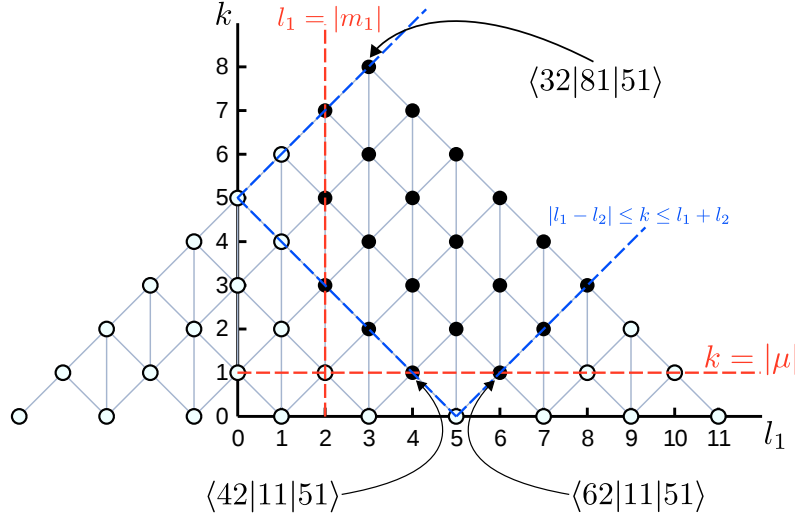


Figure 7.3.: Diagram for computing the Gaunt coefficient $\langle 32|81|51 \rangle$. All empty circles indicate zero elements. Solid circles are the non-trivial Gaunt coefficients to compute. The blue dashed lines enclose the region where the selection rules are fulfilled. The red dashed lines mark the boundaries where $l = |m|$ for Y_{lm} . To reach the target $\langle 32|81|51 \rangle$, we start from the base cases $\langle 42|11|51 \rangle$ and $\langle 62|11|51 \rangle$, then travel along the grid towards $\langle 32|81|51 \rangle$.

For the base cases, it is sufficient to consider $\langle l_1 m_1 | k k | l_2 m_2 \rangle$, since we have the inverse symmetry

$$\langle l_1 m_1 | k k | l_2 m_2 \rangle = \langle l_1, -m_1 | k, -k | l_2, -m_2 \rangle$$

For this case, we use another recursion

$$\langle l_1 m_1 | k k | l_2 m_2 \rangle = D \langle l_1 - 1, m_1 - 1 | k - 1, k - 1 | l_2 m_2 \rangle + E \langle l_1 + 1, m_1 - 1 | k - 1, k - 1 | l_2 m_2 \rangle \quad (7.11)$$

where,

$$D = \sqrt{\frac{(2k+1)(l_1+m_1-1)(l_1+m_1)}{2k(2l_1+1)(2l_1-1)}}; \quad E = -\sqrt{\frac{(2k+1)(l_1-m_1+1)(l_1-m_1+2)}{2k(2l_1+1)(2l_1+3)}}$$

This recursion iterates $Y_{kk}, Y_{k-1,k-1}, \dots, Y_{00}$. Eventually, the only non-trivial root that contributes is: $\langle l_2 m_2 | 00 | l_2 m_2 \rangle = \frac{1}{\sqrt{4\pi}}$. As an example, we illustrate the recursion diagram for computing $\langle 33|88|5, -5 \rangle$ in Fig. 7.4.

In either Fig. 7.3 or Fig. 7.4, if we fix all the other quantum numbers and vary l_1 (moving the target horizontally), the amount of computation scales linearly (reaches maximum when $l_1 = l_2$). If we fix all the other quantum numbers and vary k (moving the target vertically), the amount of computation scales quadratically (reaches maximum when $k = l_1 + l_2$). As a remark, both recursions (7.10) and (7.11) should not be implemented as a tree traversal. As we have discussed already, we can always start from the base cases and traverse the rectangular grid. This should be programmed as a simple iteration (Algorithm 7.3), which is the key of bringing exponential time scaling down to at most quadratic.

7. Recursive computation of Gaunt coefficients

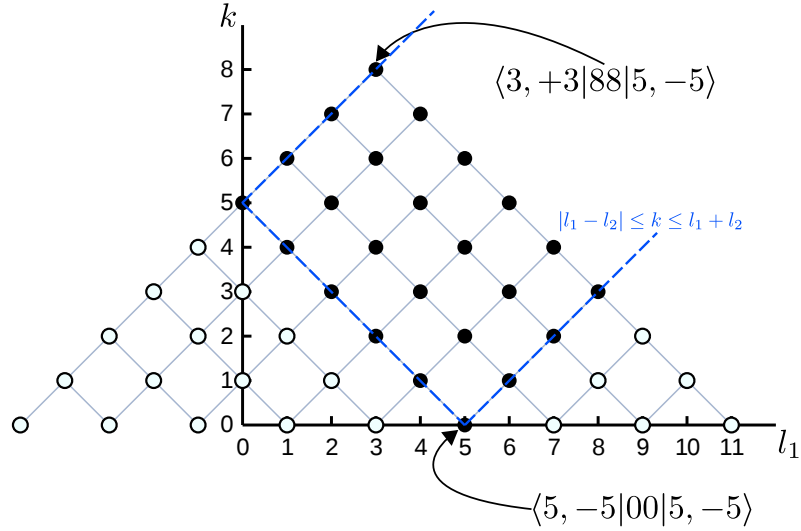


Figure 7.4.: Diagram for computing a Gaunt coefficient with $k = \mu$. To reach the Gaunt coefficient $\langle l_1 m_1 | k k | l_2 m_2 \rangle$, the starting point is: $\langle l_2 m_2 | 00 | l_2 m_2 \rangle = \frac{1}{\sqrt{4\pi}}$.

Derivation of recursion (7.10):

We start from the relation between Y_{lm} and P_l^m ,

$$Y_{lm}(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} e^{im\phi} P_l^m(\cos \theta) \quad (7.12)$$

Now consider the recursion relation of the associated Legendre polynomials,

$$(l-m+1)P_{l+1}^m = (2l+1)xP_l^m - (l+m)P_{l-1}^m \quad (7.13)$$

or, in terms of Y_{lm} ,

$$(l-m+1)\sqrt{\frac{(2l+1)(l+m+1)}{(2l+3)(l-m+1)}}Y_{l+1,m} = (2l+1)xY_{lm} - (l+m)\sqrt{\frac{(2l+1)(l-m)}{(2l-1)(l+m)}}Y_{l-1,m} \quad (7.14)$$

Now, substitutions,

$$\begin{aligned}
& \langle l_1 m_1 | \boxed{Y_{k\mu}} | l_2 m_2 \rangle \\
&= \langle l_1 m_1 | \frac{1}{k-\mu} \sqrt{\frac{(2k+1)(k-\mu)}{(2k-1)(k+\mu)}} \left[(2k-1)xY_{k-1,\mu} - (k+\mu-1) \sqrt{\frac{(2k-1)(k-\mu-1)}{(2k-3)(k+\mu-1)}} Y_{k-2,\mu} \right] | l_2 m_2 \rangle \\
&= \sqrt{\frac{(2k+1)(2k-1)}{(k+\mu)(k-\mu)}} \langle \boxed{xY_{l_1 m_1}} | k-1, \mu | l_2 m_2 \rangle - \sqrt{\frac{(2k+1)(k+\mu-1)(k-\mu-1)}{(k+\mu)(k-\mu)(2k-3)}} \langle l_1 m_1 | k-2, \mu | l_2 m_2 \rangle \\
&= \sqrt{\frac{(2k+1)(2k-1)}{(k+\mu)(k-\mu)}} \left[\frac{l_1+m_1}{2l_1+1} \sqrt{\frac{(2l_1+1)(l_1-m_1)}{(2l_1-1)(l_1+m_1)}} \langle l_1-1, m_1 | k-1, \mu | l_2 m_2 \rangle \right. \\
&\quad \left. + \frac{l_1-m_1+1}{2l_1+1} \sqrt{\frac{(2l_1+1)(l_1+m_1+1)}{(2l_1+3)(l_1-m_1+1)}} \langle l_1+1, m_1 | k-1, \mu | l_2 m_2 \rangle \right] \\
&\quad - \sqrt{\frac{(2k+1)(k+\mu-1)(k-\mu-1)}{(k+\mu)(k-\mu)(2k-3)}} \langle l_1 m_1 | k-2, \mu | l_2 m_2 \rangle \\
&= \sqrt{\frac{(2k+1)(2k-1)(l_1+m_1)(l_1-m_1)}{(k+\mu)(k-\mu)(2l_1+1)(2l_1-1)}} \langle l_1-1, m_1 | k-1, \mu | l_2 m_2 \rangle \\
&\quad + \sqrt{\frac{(2k+1)(2k-1)(l_1+m_1+1)(l_1-m_1+1)}{(k+\mu)(k-\mu)(2l_1+3)(2l_1+1)}} \langle l_1+1, m_1 | k-1, \mu | l_2 m_2 \rangle \\
&\quad - \sqrt{\frac{(2k+1)(k+\mu-1)(k-\mu-1)}{(k+\mu)(k-\mu)(2k-3)}} \langle l_1 m_1 | k-2, \mu | l_2 m_2 \rangle \quad \text{Q.E.D.} \tag{7.15}
\end{aligned}$$

Derivation of recursion (7.11):

Consider the recursion for P_l^l :

$$P_l^l = -(2l-1)\sqrt{1-x^2}P_{l-1}^{l-1} \tag{7.16}$$

In terms of Y_{kk} , we have

$$Y_{kk} = -\sqrt{\frac{2k+1}{2k}} \sqrt{1-x^2} e^{i\phi} Y_{k-1,k-1} \tag{7.17}$$

Now,

$$\begin{aligned}
\langle l_1 m_1 | \boxed{Y_{kk}} | l_2 m_2 \rangle &= -\langle l_1 m_1 | \sqrt{\frac{2k+1}{2k}} \sqrt{1-x^2} e^{i\phi} Y_{k-1,k-1} | l_2 m_2 \rangle \\
&= -\sqrt{\frac{2k+1}{2k}} \langle \sqrt{1-x^2} e^{-i\phi} Y_{l_1 m_1} | k-1, k-1 | l_2 m_2 \rangle \tag{7.18}
\end{aligned}$$

Consider another recursion for P_l^m :

$$\sqrt{1-x^2}P_l^m = \frac{1}{2l+1} \left[(l-m+1)(l-m+2)P_{l+1}^{m-1} - (l+m-1)(l+m)P_{l-1}^{m-1} \right] \tag{7.19}$$

7. Recursive computation of Gaunt coefficients

In terms of Y_{lm} , we have

$$\sqrt{1-x^2}e^{-i\phi}Y_{lm} = \sqrt{\frac{(l-m+1)(l-m+2)}{(2l+1)(2l+3)}}Y_{l+1,m-1} - \sqrt{\frac{(l+m-1)(l+m)}{(2l+1)(2l-1)}}Y_{l-1,m-1} \quad (7.20)$$

Hence, recursion (7.18) continues:

$$\begin{aligned} & \langle l_1 m_1 | k k | l_2 m_2 \rangle \\ &= -\sqrt{\frac{2k+1}{2k}} \langle \sqrt{1-x^2}e^{-i\phi}Y_{l_1 m_1} | k-1, k-1 | l_2 m_2 \rangle \\ &= \sqrt{\frac{(2k+1)(l_1+m_1-1)(l_1+m_1)}{2k(2l_1+1)(2l_1-1)}} \langle l_1-1, m_1-1 | k-1, k-1 | l_2 m_2 \rangle \\ &\quad - \sqrt{\frac{(2k+1)(l_1-m_1+1)(l_1-m_1+2)}{2k(2l_1+1)(2l_1+3)}} \langle l_1+1, m_1-1 | k-1, k-1 | l_2 m_2 \rangle \quad \text{Q.E.D.} \quad (7.21) \end{aligned}$$

7.6. Re-ordering

Gaunt coefficients are highly oscillating (see Fig. 7.2), numerically it is a big challenge to compute them using finite-precision arithmetic. Here we will discuss a simple re-ordering technique that can significantly improve the numerical accuracy.

It should be clear that, for instance, the following two Gaunt coefficients are identical,

$$\int d\Omega \overline{Y_{l_1 m_1}} Y_{k\mu} Y_{l_2 m_2} = \int d\Omega \overline{Y_{l_1 m_1}} Y_{l_2 m_2} Y_{k\mu}$$

since we only re-ordered two functions in the integrand. Using the relation,

$$\overline{Y_{lm}} = (-1)^m Y_{l,-m} \quad (7.22)$$

we can see that the following 6 Gaunt coefficients with re-ordered spherical harmonics are equivalent:

$$\begin{aligned} \langle l_1 m_1 | k\mu | l_2 m_2 \rangle & \quad (-1)^{m_2} \langle k, -\mu | l_1, -m_1 | l_2 m_2 \rangle & \quad (-1)^\mu \langle l_2, -m_2 | l_1, -m_1 | k\mu \rangle \\ \langle l_1 m_1 | l_2 m_2 | k\mu \rangle & \quad (-1)^{m_2} \langle k, -\mu | l_2 m_2 | l_1, -m_1 \rangle & \quad (-1)^\mu \langle l_2, -m_2 | k\mu | l_1, -m_1 \rangle \end{aligned}$$

Although these elements are analytically identical, our recursion relation treats l_1 , l_2 , and k differently. Numerically, they can produce different results due to finite-precision arithmetic. An example of numerical errors for producing a Gaunt matrix $\langle 30m_1 | 40\mu | 30m_2 \rangle$ is shown in Fig. 7.5. The errors are plotted in the $\log_{10}$ scale and color mapped, where the dark blue represents error less than 10^{-16} and the bright yellow represents error greater than 10^{-10} . The error patterns look differently with the different choices of ordering.

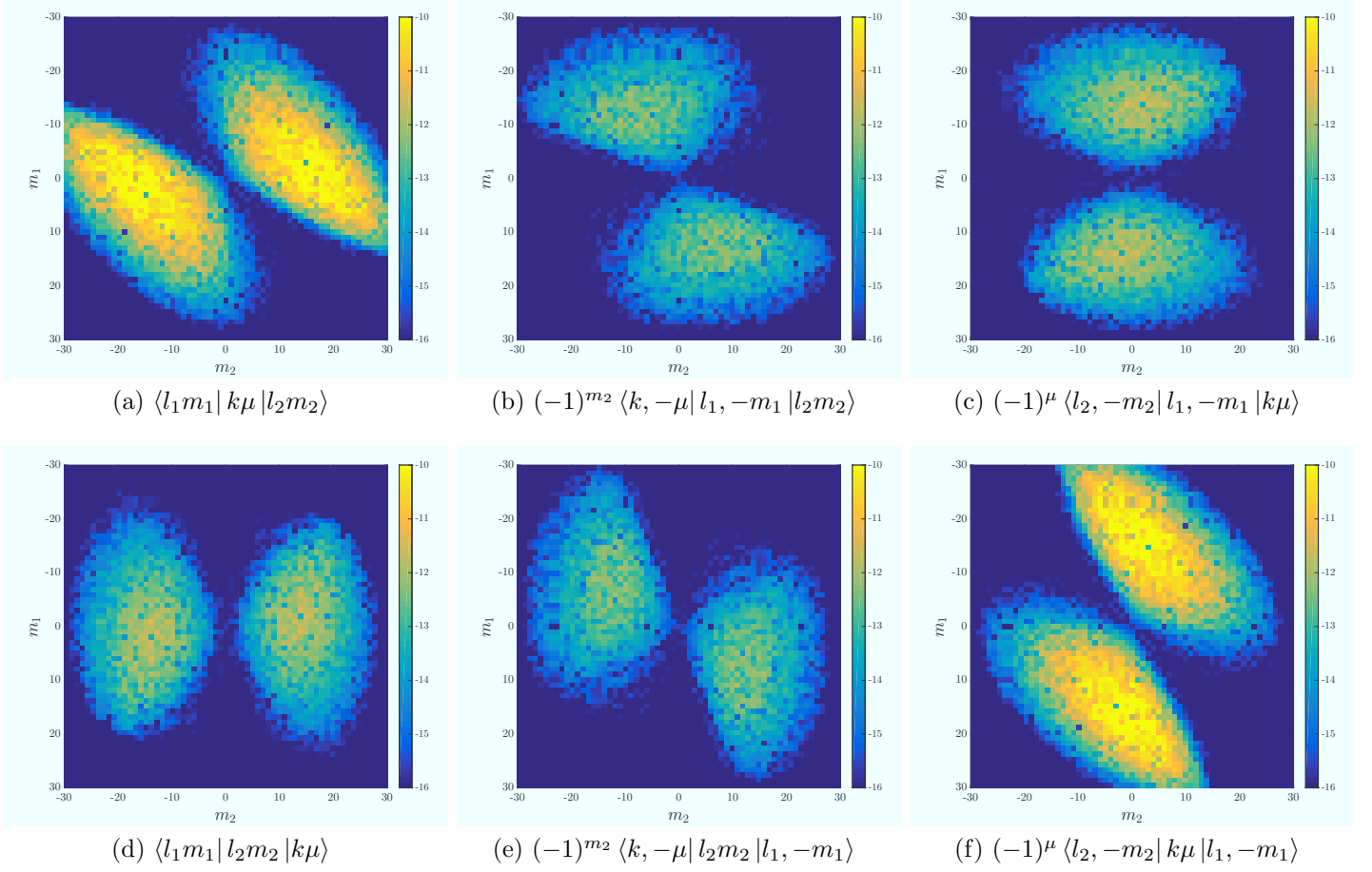


Figure 7.5.: Error of Gaunt coefficients ($l_1 = l_2 = 30, k = 40$) computed with different ordering. Dark blue: elements with small error; Bright yellow: elements with large error.

7. Recursive computation of Gaunt coefficients

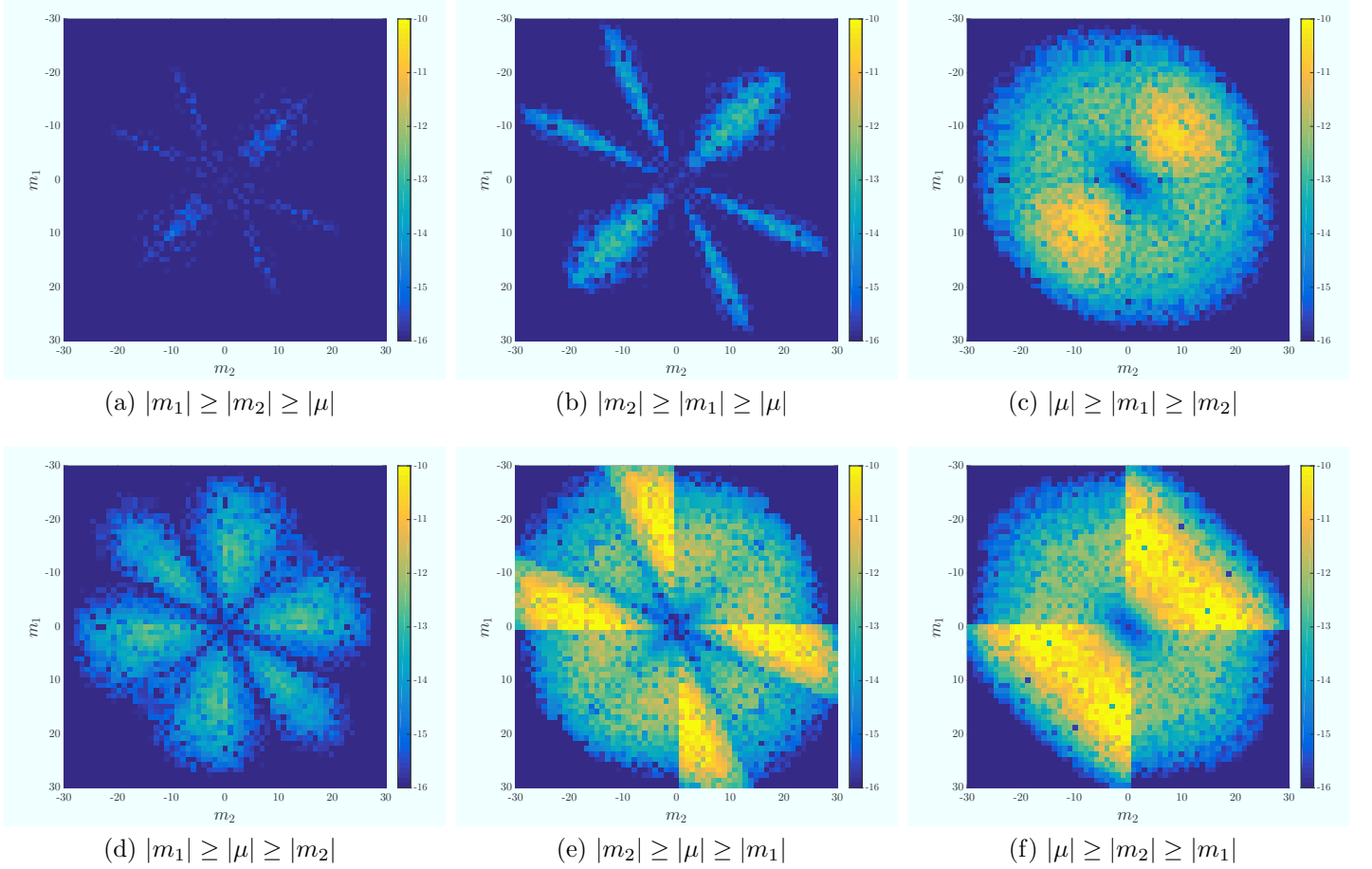


Figure 7.6.: Error of Gaunt coefficients ($l_1 = l_2 = 30$, $k = 40$) computed with different ordering. Dark blue: elements with small error; Bright yellow: elements with large error.

If we overlap the 6 error images in Fig. 7.5, the minimum error at each element tells the lowest possible error in principle we can achieve using our recursion. The question is, how should we re-order the three spherical harmonics to achieve the minimum error? We have tested 6 options of sorting $|m_1|$, $|m_2|$, and $|\mu|$ element-wise. The results are shown in Fig. 7.6. Evidently, the re-ordering with $|m_1| \geq |m_2| \geq |\mu|$ produces the lowest error. The rationale behind is that (consider $l_1 = l_2 = k$ for simplicity) when we maximize $|m_1|$, we are pushing the red vertical line in Fig. 7.3 to the right as much as possible, thus reducing the number of Gaunt coefficients to be computed; when we minimize $|\mu|$, we are lowering the red horizontal line in Fig. 7.3 as much as possible, thus minimizing the k value in Y_{kk} of the base cases, as error on base cases increases with k .

Algorithm 7.3: Compute Gaunt coefficients using the recursion relations (`gaunt.py`).

```

1  import math
2  import numpy as np
3
4  # Recursion coefficients
5  def A(l, k, m, mu):
6      return math.sqrt((2*k+1)*(2*k-1)*(l+m)*(l-m)/((k+mu)*(k-mu)*(2*l+1)*(2*l-1)))
7  def B(l, k, m, mu):
8      return math.sqrt((2*k+1)*(2*k-1)*(l+m+1)*(l-m+1)/((k+mu)*(k-mu)*(2*l+3)*(2*l+1)))
9  def C(k, mu):
10     return -math.sqrt((2*k+1)*(k+mu-1)*(k-mu-1)/((k+mu)*(k-mu)*(2*k-3)))
11 def D(l, k, m):
12     return math.sqrt((2*k+1)*(l+m-1)*(l+m)/(2*k*(2*l+1)*(2*l-1)))
13 def E(l, k, m):
14     return -math.sqrt((2*k+1)*(l-m+1)*(l-m+2)/(2*k*(2*l+1)*(2*l+3)))
15
16 # Base case <l1m1|kk|l2m2>*sqrt(4*pi)
17 def Base(l1, l2, k, m1, m2):
18     # Trivial cases
19     if l1<abs(m1) or l2<abs(m2) or m1-m2!=k or (l1+l2+k)%2!=0 or not abs(l1-l2)<=k<=l1+l2:
20         return 0
21
22     # Grid size
23     (m, n) = ((k+l1-l2)//2+1, (k-l1+l2)//2+1)
24
25     # Index generator
26     idx = ((i, j) for i in range(m) for j in range(n) if not i==j==0)
27
28     # Gaunts
29     G = np.zeros((m+1, n+1)) # with last ghost row & col
30     G[0, 0] = 1.0
31     for (i, j) in idx:
32         G[i, j] = D(l2+i-j, i+j, m1-k+i+j)*G[i-1, j] \
33             + E(l2+i-j, i+j, m1-k+i+j)*G[i, j-1]
34
35     return G[m-1, n-1]
36
37 # Gaunt coefficient <l1m1|kmu|l2m2>
38 def Gaunt(l1, l2, k, m1, m2):
39     mu = m1-m2
40     # Trivial cases
41     if l1<abs(m1) or l2<abs(m2) or k<abs(mu) or (l1+l2+k)%2!=0 or not abs(l1-l2)<=k<=l1+l2:
42         return 0
43
44     # Re-arrange such that |m1|>=|m2|>=|mu|
45     sign = 1
46     if abs(m1)<abs(m2): (l1, m1, l2, m2) = (l2, -m2, l1, -m1); sign *= (-1)**mu
47     if abs(m1)<abs(mu): (l1, m1, k, mu) = (k, -mu, l1, -m1); sign *= (-1)**m2
48     if abs(m2)<abs(mu): (l2, m2, k, mu) = (k, mu, l2, m2)

```

7. Recursive computation of Gaunt coefficients

```

49     if mu<0: (m1, m2, mu) = (-m1, -m2, -mu) # Make mu positive
50
51     # Grid size
52     (m, n) = ((k+l1-l2)//2+1, (k-l1+l2)//2+1)
53
54     # Index generator
55     idx0 = ((i, j) for i in range(m) for j in range(n) if l2+i-j>=abs(m1) and i+j==abs(mu))
56     idx = ((i, j) for i in range(m) for j in range(n) if l2+i-j>=abs(m1) and i+j> abs(mu))
57
58     # Gaunts
59     G = np.zeros((m+1, n+1)) # with last ghost row & col
60     # Base cases
61     for (i, j) in idx0:
62         G[i, j] = Base(l2+i-j, l2, i+j, m1, m2)
63     # Non-base cases
64     for (i, j) in idx:
65         G[i, j] = A(l2+i-j, i+j, m1, mu)*G[i-1, j] \
66                 + B(l2+i-j, i+j, m1, mu)*G[i, j-1] \
67                 + C(i+j, mu)*G[i-1, j-1]
68
69     return sign*G[m-1, n-1]/math.sqrt(4*math.pi)

```

7.7. Error comparison among different algorithms

In 1998, Didier Sébilleau reviewed several different methods for evaluating Gaunt coefficients [50]. Sébilleau kindly provided me the source codes which were implemented in [50]. Among the several algorithms in [50], the most successful one was the Schulten-Gordon method, which is based on the explicit formula (7.9) and using the recursion relations among the Wigner $3j$ symbols.

In this section, we compare the error behaviors among the three methods: (a) the explicit implementation (Algorithm 7.1); (b) the Schulten-Gordon method [50]; (c) the recursive method implemented in this work (Algorithm 7.3). All the calculations are based on double-precision floating-point format. Table 7.1 lists a few selected values of Gaunt coefficients computed from the three methods. Fig. 7.7 and Fig. 7.8 show two examples of the error patterns of Gaunt matrices with $(l_1 = 30, l_2 = 20, k = 10)$ and $(l_1 = 30, l_2 = 30, k = 40)$, respectively. Fig. 7.9 plots the maximum absolute errors (in $\log_{10}$ scale) of all Gaunt coefficients computed with different methods. At $l_{\max} = 65$, the recursive method in this work produces Gaunt coefficients accurate to about 11 decimal points.

Table 7.1.: Examples of values of Gaunt coefficients.

Algorithms	$\langle l_1 m_1 k \mu l_2 m_2 \rangle$											
	l_1	k	l_2	l_1	k	l_2	l_1	k	l_2	l_1	k	l_2
	60	40	60	60	40	60	30	30	30	30	30	30
	m_1	μ	m_2	m_1	μ	m_2	m_1	μ	m_2	m_1	μ	m_2
	20	-40	60	0	0	0	30	30	0	0	0	0
Explicit	0.0000000100281641			0.0423469833493180			-0.0001173709420163			0.0530907569972774		
Schulten-Gordon	0.0000000096273087			0.0423472810278219			-0.0001173709419646			0.0530907570759748		
This work	0.0000000100281641			0.0423472810278134			-0.0001173709420162			0.0530907570759795		
Exact	0.0000000100281641			0.0423472810278135			-0.0001173709420163			0.0530907570759797		

7.7. Error comparison among different algorithms

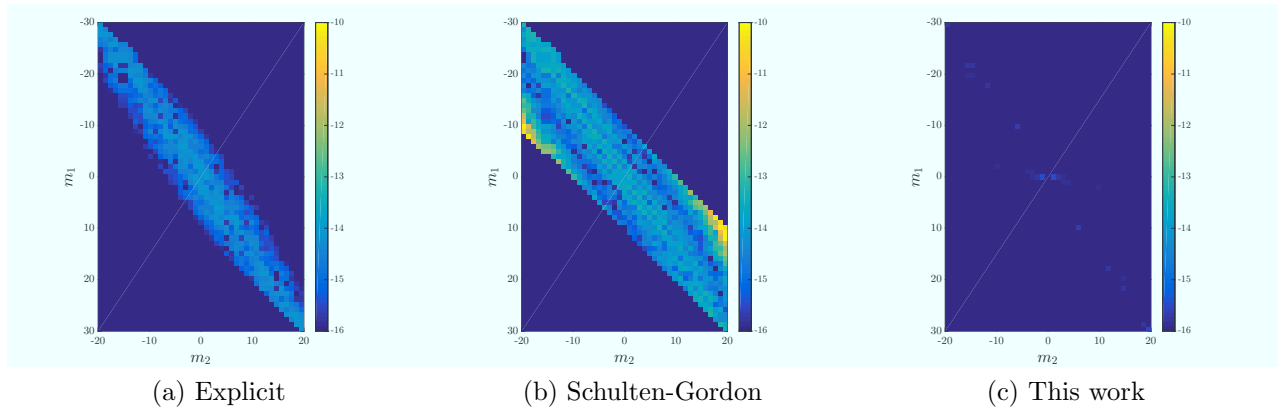


Figure 7.7.: Error patterns of Gaunt matrices with $l_1 = 30$, $l_2 = 20$, and $k = 10$. (a) Explicit formula; (b) Schulten-Gordon; (c) This work.

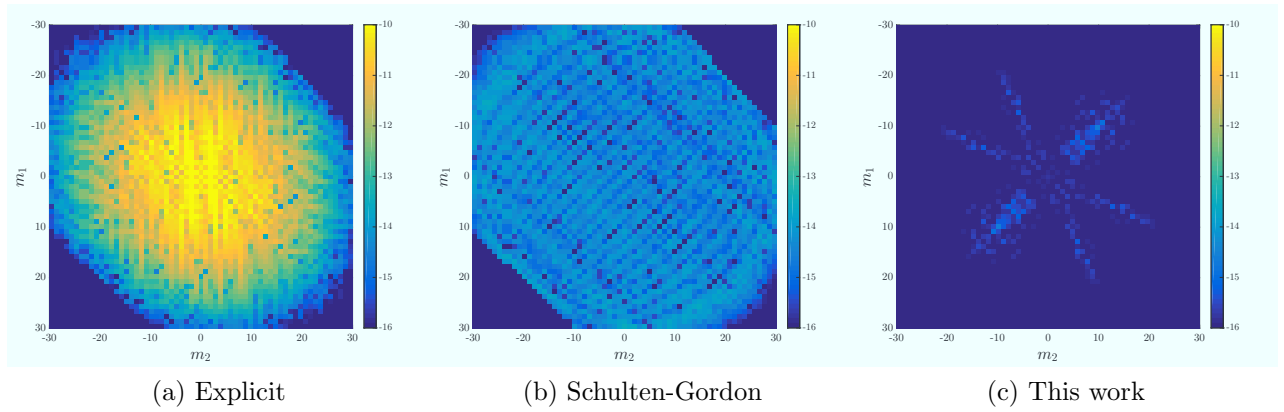


Figure 7.8.: Error patterns of Gaunt matrices with $l_1 = l_2 = 30$ and $k = 40$. (a) Explicit formula; (b) Schulten-Gordon; (c) This work.

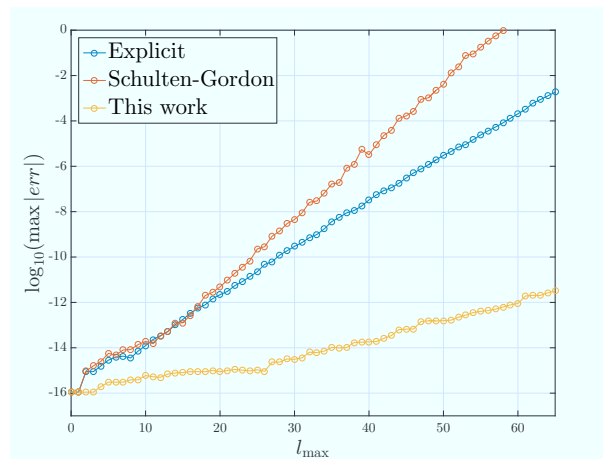


Figure 7.9.: Maximum absolute errors (in $\log_{10}$ scale) of all Gaunt coefficients computed with different methods using double-precision floating-point format.

7.8. List of the first few Gaunt coefficients in matrix form

$$\langle 0m_1 | 0\mu | 0m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \begin{bmatrix} 1 \end{bmatrix} \quad (7.23)$$

$$\langle 1m_1 | 0\mu | 1m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.24)$$

$$\langle 1m_1 | 2\mu | 1m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \frac{\sqrt{5}}{5} \begin{bmatrix} -1 & \sqrt{3} & -\sqrt{6} \\ -\sqrt{3} & 2 & -\sqrt{3} \\ -\sqrt{6} & \sqrt{3} & -1 \end{bmatrix} \quad (7.25)$$

$$\langle 2m_1 | 0\mu | 2m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.26)$$

$$\langle 2m_1 | 2\mu | 2m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \frac{\sqrt{5}}{7} \begin{bmatrix} -2 & \sqrt{6} & -2 & 0 & 0 \\ -\sqrt{6} & 1 & 1 & -\sqrt{6} & 0 \\ -2 & -1 & 2 & -1 & -2 \\ 0 & -\sqrt{6} & 1 & 1 & -\sqrt{6} \\ 0 & 0 & -2 & \sqrt{6} & -2 \end{bmatrix} \quad (7.27)$$

$$\langle 2m_1 | 4\mu | 2m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \frac{1}{7} \begin{bmatrix} 1 & -\sqrt{5} & \sqrt{15} & -\sqrt{35} & \sqrt{70} \\ \sqrt{5} & -4 & \sqrt{30} & -\sqrt{40} & \sqrt{35} \\ \sqrt{15} & -\sqrt{30} & 6 & -\sqrt{30} & \sqrt{15} \\ \sqrt{35} & -\sqrt{40} & \sqrt{30} & -4 & \sqrt{5} \\ \sqrt{70} & -\sqrt{35} & \sqrt{15} & -\sqrt{5} & 1 \end{bmatrix} \quad (7.28)$$

7.8. List of the first few Gaunt coefficients in matrix form

$$\langle 3m_1 | 0\mu | 3m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.29)$$

$$\langle 3m_1 | 2\mu | 3m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \frac{\sqrt{5}}{15} \begin{bmatrix} -5 & 5 & -\sqrt{10} & 0 & 0 & 0 & 0 \\ -5 & 0 & \sqrt{15} & -\sqrt{20} & 0 & 0 & 0 \\ -\sqrt{10} & -\sqrt{15} & 3 & \sqrt{2} & -\sqrt{24} & 0 & 0 \\ 0 & -\sqrt{20} & -\sqrt{2} & 4 & -\sqrt{2} & -\sqrt{20} & 0 \\ 0 & 0 & -\sqrt{24} & \sqrt{2} & 3 & -\sqrt{15} & -\sqrt{10} \\ 0 & 0 & 0 & -\sqrt{20} & \sqrt{15} & 0 & -5 \\ 0 & 0 & 0 & 0 & -\sqrt{10} & 5 & -5 \end{bmatrix} \quad (7.30)$$

$$\langle 3m_1 | 4\mu | 3m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \frac{1}{11} \begin{bmatrix} 3 & -\sqrt{30} & \sqrt{54} & -\sqrt{63} & \sqrt{42} & 0 & 0 \\ \sqrt{30} & -7 & \sqrt{32} & -\sqrt{3} & -\sqrt{14} & \sqrt{70} & 0 \\ \sqrt{54} & -\sqrt{32} & 1 & \sqrt{15} & -\sqrt{40} & \sqrt{14} & \sqrt{42} \\ \sqrt{63} & -\sqrt{3} & -\sqrt{15} & 6 & -\sqrt{15} & -\sqrt{3} & \sqrt{63} \\ \sqrt{42} & \sqrt{14} & -\sqrt{40} & \sqrt{15} & 1 & -\sqrt{32} & \sqrt{54} \\ 0 & \sqrt{70} & -\sqrt{14} & -\sqrt{3} & \sqrt{32} & -7 & \sqrt{30} \\ 0 & 0 & \sqrt{42} & -\sqrt{63} & \sqrt{54} & -\sqrt{30} & 3 \end{bmatrix} \quad (7.31)$$

$$\langle 3m_1 | 6\mu | 3m_2 \rangle:$$

$$\frac{1}{\sqrt{4\pi}} \frac{5\sqrt{13}}{429} \begin{bmatrix} -1 & \sqrt{7} & -\sqrt{28} & \sqrt{84} & -\sqrt{210} & \sqrt{462} & -\sqrt{924} \\ -\sqrt{7} & 6 & -\sqrt{105} & \sqrt{224} & -\sqrt{378} & \sqrt{504} & -\sqrt{462} \\ -\sqrt{28} & \sqrt{105} & -15 & \sqrt{350} & -\sqrt{420} & \sqrt{378} & -\sqrt{210} \\ -\sqrt{84} & \sqrt{224} & -\sqrt{350} & 20 & -\sqrt{350} & \sqrt{224} & -\sqrt{84} \\ -\sqrt{210} & \sqrt{378} & -\sqrt{420} & \sqrt{350} & -15 & \sqrt{105} & -\sqrt{28} \\ -\sqrt{462} & \sqrt{504} & -\sqrt{378} & \sqrt{224} & -\sqrt{105} & 6 & -\sqrt{7} \\ -\sqrt{924} & \sqrt{462} & -\sqrt{210} & \sqrt{84} & -\sqrt{28} & \sqrt{7} & -1 \end{bmatrix} \quad (7.32)$$

8. Orbital overlaps and basis orthonormalization

8.1. Orbital overlaps

Imagine the simple picture (Fig. 8.1): a $1s$ orbital $\varphi_{1s}(\mathbf{r})$ is centered around the origin, another $1s$ orbital $\varphi_{1s}(\mathbf{r}-\mathbf{a})$ is centered about position $\mathbf{a}$.

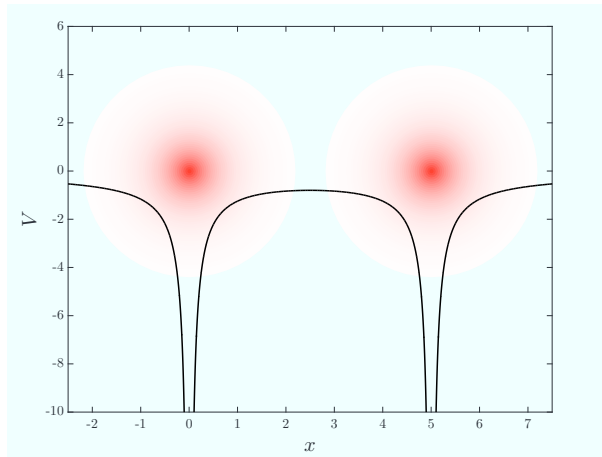


Figure 8.1.: A two-site system with two $1s$ orbitals. This figure only serves as showing a basic concept, as we are kind of drawing 3D wave functions on a 1D potential.

If the two sites are well separated (significantly greater than a Bohr radius), to a good approximation, we could ignore the overlap and treat the two orbitals as orthonormal. But if the two sites are close, it is necessary to evaluate the overlap

$$\langle \varphi_{1s}(\mathbf{r}) | \varphi_{1s}(\mathbf{r}-\mathbf{a}) \rangle = \int d\mathbf{r} \overline{\varphi_{1s}(\mathbf{r})} \varphi_{1s}(\mathbf{r}-\mathbf{a}) \quad (8.1)$$

Evaluating such an integral is not a trivial task (especially for orbitals with higher angular momenta), as it involves “two centers”. Without unnecessary excursions, I directly introduce the spirit of conducting such integrations. It consists of two steps:

1. Expand each atomic orbital into plane waves;
2. Integrate over the plane waves.

8. Orbital overlaps and basis orthonormalization

At first, this might seem crazy: an atomic orbital does not look like a plane wave at all! One must use *a lot of* plane waves to construct an atomic orbital. Expanding an arbitrary function into plane waves is nothing but the Fourier transformation. As we are going to explain step by step, this expansion from an atomic orbital into plane waves won't be too difficult. But first of all, why plane waves? What is so good about plane waves? The answer is: they are free to translate. Translating a plane wave is nothing but multiplying by a phase prefactor. Thus we can easily get rid of the two-center difficulty.

Consider again (8.1): Suppose we have the two orbitals expanded in plane waves:

$$|\varphi_{1s}(\mathbf{r})\rangle = \int d\mathbf{k} c(\mathbf{k}) |\mathbf{k}\rangle \quad (8.2)$$

$$|\varphi_{1s}(\mathbf{r}-\mathbf{a})\rangle = \int d\mathbf{k} e^{-i\mathbf{k}\cdot\mathbf{a}} c(\mathbf{k}) |\mathbf{k}\rangle \quad (8.3)$$

where $c(\mathbf{k})$'s are the expansion coefficients. Now,

$$\begin{aligned} \langle \varphi_{1s}(\mathbf{r}) | \varphi_{1s}(\mathbf{r}-\mathbf{a}) \rangle &= \int d\mathbf{k}_1 \int d\mathbf{k}_2 \bar{c}(\mathbf{k}_1) c(\mathbf{k}_2) e^{-i\mathbf{k}_2\cdot\mathbf{a}} \underbrace{\langle \mathbf{k}_1 | \mathbf{k}_2 \rangle}_{\delta(\mathbf{k}_1 - \mathbf{k}_2)} \\ &= \int d\mathbf{k} \bar{c}(\mathbf{k}) c(\mathbf{k}) e^{-i\mathbf{k}\cdot\mathbf{a}} \end{aligned} \quad (8.4)$$

Integrating (8.4) is much easier than the original (8.1). Apparently, the most urgent question is: how to expand an atomic orbital into plane waves?

8.2. Plane waves and spherical waves

Expanding an atomic orbital into plane waves consists of two steps:

1. Expand the atomic orbital into spherical waves;
2. Expand each spherical wave into plane waves.

First of all, it would be nice to have a brief overview on the concepts of plane waves and spherical waves. Remember that they are **both** solutions of the free particle Schrödinger equation, while the plane waves are solutions obtained in Cartesian coordinates ($\varphi(\mathbf{r}) = X(x)Y(y)Z(z)$) and the spherical waves are solutions obtained in spherical coordinates ($\varphi(\mathbf{r}) = R(r)Y(\theta, \phi)$).

	Plane wave	Spherical wave
Dirac notation	$ \mathbf{k}\rangle$	$ klm\rangle$
Real space	$\frac{1}{(\sqrt{2\pi})^3} e^{i\mathbf{k}\cdot\mathbf{r}}$	$\sqrt{\frac{2}{\pi}} k j_l(kr) Y_{lm}(\theta, \phi)$

In real space representation, the prefactors

$$\frac{1}{(\sqrt{2\pi})^3} \quad \text{and} \quad \sqrt{\frac{2}{\pi}}k$$

are chosen such that the waves are orthonormal:

$$\langle \mathbf{k}_1 | \mathbf{k}_2 \rangle = \delta(\mathbf{k}_1 - \mathbf{k}_2) \quad (8.5)$$

$$\langle k_1 l_1 m_1 | k_2 l_2 m_2 \rangle = \delta(k_1 - k_2) \delta_{l_1 l_2} \delta_{m_1 m_2} \quad (8.6)$$

Note that neither a plane wave nor a spherical wave is actually “normalized” (as they cannot be): $\langle \mathbf{k} | \mathbf{k} \rangle = \infty$ and $\langle klm | klm \rangle = \infty$. Unlike a discrete basis set, the plane waves and the spherical waves form continuous basis sets, which are orthonormalized “Diracly”.

The orthonormality enables us to have the completeness relations

$$\int d\mathbf{k} |\mathbf{k}\rangle \langle \mathbf{k}| = \mathbb{1} \quad (8.7)$$

$$\int_0^\infty dk \sum_{l=0}^\infty \sum_{m=-l}^l |klm\rangle \langle klm| = \mathbb{1} \quad (8.8)$$

The task is to expand

$$|\varphi_\alpha\rangle = \int d\mathbf{k} |\mathbf{k}\rangle \langle \mathbf{k}| \varphi_\alpha\rangle \quad (8.9)$$

It seems like the spherical waves not involved. But evaluating the coefficient $\langle \mathbf{k} | \varphi_\alpha \rangle$ requires an intermediate step of the spherical wave expansion.

8.2.1. Expand a plane wave into spherical waves

It is important to get used to how the waves can be transformed in terms of each other. Our first task is to expand a specific plane wave in terms of spherical waves.

$$|\mathbf{k}\rangle = \int_0^\infty dk' \sum_{l'=0}^\infty \sum_{m=-l}^l |k'lm\rangle \langle k'lm | \mathbf{k}\rangle \quad (8.10)$$

Of course, the most difficult part is to work out the coefficient,

$$\langle k'lm | \mathbf{k}\rangle = \frac{i^l}{k} \overline{Y_{lm}(\hat{\mathbf{k}})} \delta(k' - k) \quad (8.11)$$

which leads to,

$$\boxed{|\mathbf{k}\rangle = \sum_{l=0}^\infty \sum_{m=-l}^l \frac{i^l}{k} \overline{Y_{lm}(\hat{\mathbf{k}})} |klm\rangle} \quad (8.12)$$

8. Orbital overlaps and basis orthonormalization

Eqn. (8.11) can be verified by bringing (8.12) back to real space, which recovers the plane wave expansion [54] (DLMF 10.60.7)

$$\begin{aligned} \frac{1}{(\sqrt{2\pi})^3} e^{i\mathbf{k}\cdot\mathbf{r}} &= \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{i^l}{k} \overline{Y_{lm}(\hat{\mathbf{k}})} \sqrt{\frac{2}{\pi}} k j_l(kr) Y_{lm}(\hat{\mathbf{r}}) \\ e^{i\mathbf{k}\cdot\mathbf{r}} &= 4\pi \sum_{l=0}^{\infty} i^l j_l(kr) \sum_{m=-l}^l \overline{Y_{lm}(\hat{\mathbf{k}})} Y_{lm}(\hat{\mathbf{r}}) \end{aligned} \quad (8.13)$$

Using the addition theorem, one normally writes (8.13) in a simpler form:

$$e^{i\mathbf{k}\cdot\mathbf{r}} = \sum_{l=0}^{\infty} (2l+1) i^l j_l(kr) P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}) \quad (8.14)$$

This is something quite interesting to think about: keep adding spherical waves, you eventually recover a plane wave! I would like to visualize this picture. Let's consider a plane wave traveling in $\hat{\mathbf{z}}$ direction.

We can not easily plot a plane wave with complex amplitude. But a linear combination of two plane waves can produce a wave with only real amplitude.

$$\cos(\mathbf{k} \cdot \mathbf{r}) = \frac{e^{i\mathbf{k}\cdot\mathbf{r}} + e^{-i\mathbf{k}\cdot\mathbf{r}}}{2} \quad (8.15)$$

Therefore,

$$\begin{aligned} \cos(\mathbf{k} \cdot \mathbf{r}) &= \frac{1}{2} \sum_{l=0}^{\infty} (2l+1) i^l j_l(kr) \left[P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}) + P_l(-\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}) \right] \\ &= \frac{1}{2} \sum_{l=0}^{\infty} (2l+1) i^l j_l(kr) \left[P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}) + (-1)^l P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}) \right] \\ &= \sum_{l=0,2,4,\dots}^{\infty} (2l+1) i^l j_l(kr) P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}) \\ &= \sum_{l=0,2,4,\dots}^{\infty} (2l+1) (-1)^{l/2} j_l(kr) P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}) \end{aligned} \quad (8.16)$$

Now we orient the wave pointing into the $\hat{\mathbf{z}}$ direction, thus $\hat{\mathbf{z}} \cdot \hat{\mathbf{r}} = \cos \theta$.

$$\cos(kr \cos \theta) = \sum_{l=0,2,4,\dots}^{\infty} (2l+1) (-1)^{l/2} j_l(kr) P_l(\cos \theta) \quad (8.17)$$

This equation can be very easily visualized. Recall the first few spherical Bessel functions:

$$j_0(z) = \frac{\sin z}{z}, \quad j_1(z) = \frac{\sin z}{z^2} - \frac{\cos z}{z}, \quad j_2(z) = \left(\frac{3}{z^3} - \frac{1}{z} \right) \sin z - \frac{3}{z^2} \cos z \quad (8.18)$$

The construction of the plane wave is shown in Fig. 8.2. As the spherical waves accumulate, the radius of the circle enclosing the well formed plane wave region expands linearly. This gives us a hint that for solving general two-center problems, the number of spherical waves required increases linearly with the distance of the two sites. For a more general discussion, see Section 9.1.

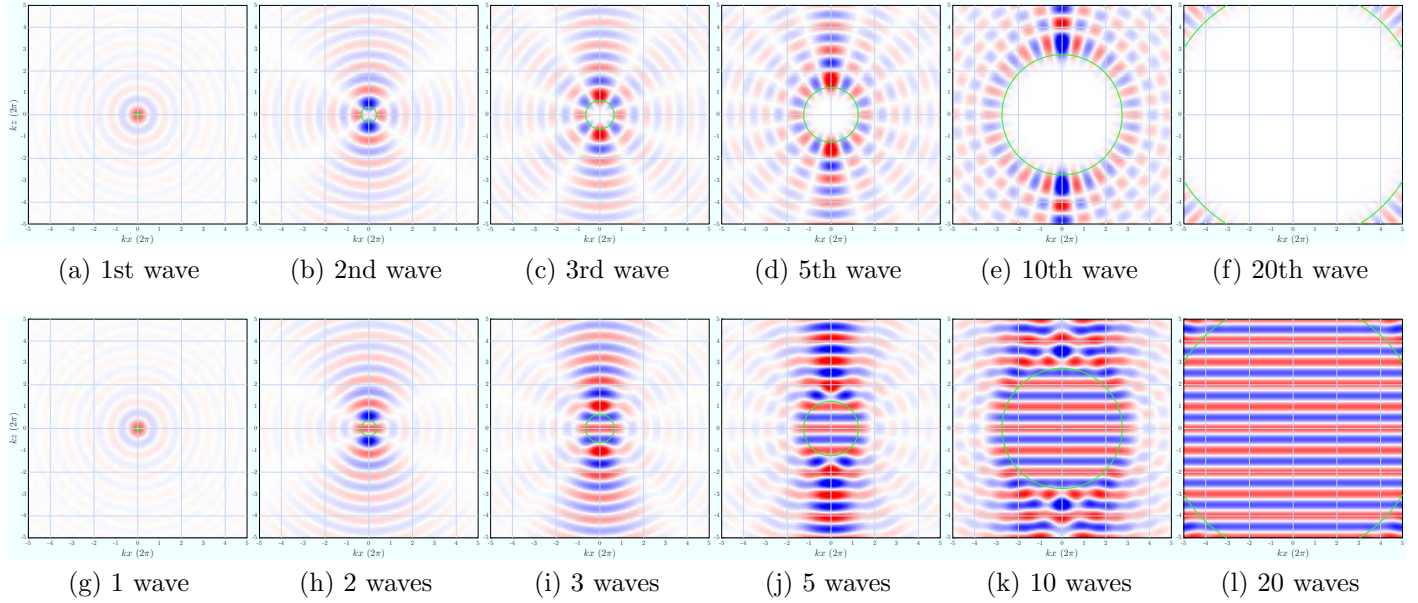


Figure 8.2.: Individual spherical waves add up to a plane wave traveling in $\hat{\mathbf{z}}$ direction. (a)-(f) individual spherical waves; (g)-(l) accumulated waves. The green circle encloses the well formed plane wave region.

8.2.2. Expand a spherical wave into plane waves

The other way around: expand a spherical wave in terms of plane waves.

$$|klm\rangle = \int d\mathbf{k}' |\mathbf{k}'\rangle \langle \mathbf{k}' | klm\rangle \quad (8.19)$$

The coefficient is the complex conjugate of (8.11):

$$\langle \mathbf{k}' | klm\rangle = \overline{\langle klm | \mathbf{k}'\rangle} = \frac{i^{-l}}{k} Y_{lm}(\hat{\mathbf{k}}) \delta(k' - k) \quad (8.20)$$

Therefore, (note: $\int d\mathbf{k} = \int dk k^2 \int d\hat{\mathbf{k}}$),

$$|klm\rangle = i^{-l} k \int d\hat{\mathbf{k}} Y_{lm}(\hat{\mathbf{k}}) |\mathbf{k}\rangle \quad (8.21)$$

8.2.3. Expand an atomic orbital into spherical waves

Expand an atomic orbital into spherical waves

$$|\varphi_{nlm}\rangle = \int_0^\infty dk \sum_{l'=0}^\infty \sum_{m'=-l'}^{l'} |kl'm'\rangle \langle kl'm' | \varphi_{nlm}\rangle \quad (8.22)$$

8. Orbital overlaps and basis orthonormalization

The coefficient has a simple form thanks to the angular part:

$$\langle kl'm' | \varphi_{nlm} \rangle = \sqrt{\frac{2}{\pi}} k \int_0^\infty dr r^2 j_{l'}(kr) R_{nl}(r) \int d\hat{\mathbf{r}} \overline{Y_{l'm'}(\hat{\mathbf{r}})} Y_{lm}(\hat{\mathbf{r}}) = k I_{nl}(k) \delta_{l'm',lm} \quad (8.23)$$

where we have defined,

$$I_{nl}(k) \equiv \sqrt{\frac{2}{\pi}} \int_0^\infty dr r^2 j_l(kr) R_{nl}(r) \quad (8.24)$$

Therefore,

$$|\varphi_{nlm}\rangle = \int_0^\infty dk k I_{nl}(k) |klm\rangle \quad (8.25)$$

8.2.4. Expand an atomic orbital into plane waves

Finally, expanding an atomic orbital into plane waves:

$$|\varphi_{nlm}\rangle = \int d\mathbf{k} |\mathbf{k}\rangle \langle \mathbf{k} | \varphi_{nlm} \rangle \quad (8.26)$$

As mentioned in the beginning, evaluating $\langle \mathbf{k} | \varphi_{nlm} \rangle$ is complicated. Thus, we first turned our atomic orbital into spherical waves (8.25). Then, for each spherical wave, we transform it into plane waves (8.21). In a compact form,

$$|\varphi_{nlm}\rangle = i^{-l} \int d\mathbf{k} I_{nl}(k) Y_{lm}(\hat{\mathbf{k}}) |\mathbf{k}\rangle \quad (8.27)$$

Table 8.1 summarizes the transformations that we have discussed.

Table 8.1.: Transformations among plane waves, spherical waves, and atomic orbitals.

$$\begin{aligned} |\mathbf{k}\rangle &= \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{i^l}{k} \overline{Y_{lm}(\hat{\mathbf{k}})} |klm\rangle \\ |klm\rangle &= i^{-l} k \int d\hat{\mathbf{k}} Y_{lm}(\hat{\mathbf{k}}) |\mathbf{k}\rangle \\ |\varphi_{nlm}\rangle &= \int_0^\infty dk k I_{nl}(k) |klm\rangle \\ |\varphi_{nlm}\rangle &= i^{-l} \int d\mathbf{k} I_{nl}(k) Y_{lm}(\hat{\mathbf{k}}) |\mathbf{k}\rangle \end{aligned}$$

8.3. Overlap formula for atomic orbitals

We introduce a simplified notation:

$$|\varphi_\alpha(\mathbf{r}-\mathbf{a})\rangle \mapsto |\alpha(\mathbf{a})\rangle \quad (8.28)$$

By default, $|\alpha\rangle$ denotes orbital- α centered at the origin.

The overlap of α centered at the origin and β centered at $\mathbf{a}$ is thus,

$$\langle\alpha|\beta(\mathbf{a})\rangle$$

Using (8.27) and (8.13), we can derive the general solution of orbital overlaps,

$$\begin{aligned} \langle\alpha|\beta(\mathbf{a})\rangle &= \left(i^{l_\alpha} \int d\mathbf{k}_1 \overline{I_\alpha}(k_1) \overline{Y_\alpha}(\hat{\mathbf{k}}_1) \langle\mathbf{k}_1| \right) \left(i^{-l_\beta} \int d\mathbf{k}_2 I_\beta(k_2) Y_\beta(\hat{\mathbf{k}}_2) e^{-i\mathbf{k}_2 \cdot \mathbf{a}} |\mathbf{k}_2\rangle \right) \\ &= i^{l_\alpha-l_\beta} \int d\mathbf{k}_1 \int d\mathbf{k}_2 \overline{I_\alpha}(k_1) I_\beta(k_2) \overline{Y_\alpha}(\hat{\mathbf{k}}_1) Y_\beta(\hat{\mathbf{k}}_2) e^{-i\mathbf{k}_2 \cdot \mathbf{a}} \underbrace{\langle\mathbf{k}_1|\mathbf{k}_2\rangle}_{\delta(\mathbf{k}_1-\mathbf{k}_2)} \\ &= i^{l_\alpha-l_\beta} \int d\mathbf{k} \overline{I_\alpha}(k) I_\beta(k) \overline{Y_\alpha}(\hat{\mathbf{k}}) Y_\beta(\hat{\mathbf{k}}) e^{-i\mathbf{k} \cdot \mathbf{a}} \\ &= i^{l_\alpha-l_\beta} \int d\mathbf{k} \overline{I_\alpha}(k) I_\beta(k) \overline{Y_\alpha}(\hat{\mathbf{k}}) Y_\beta(\hat{\mathbf{k}}) \left[4\pi \sum_{\lambda=0}^{\infty} i^{-\lambda} j_\lambda(ka) \sum_{\mu=-\lambda}^{\lambda} Y_{\lambda\mu}(\hat{\mathbf{k}}) \overline{Y_{\lambda\mu}}(\hat{\mathbf{a}}) \right] \\ &= 4\pi \sum_{\lambda=0}^{\infty} i^{l_\alpha-l_\beta-\lambda} \int_0^\infty dk k^2 \overline{I_\alpha}(k) I_\beta(k) j_\lambda(ka) \sum_{\mu=-\lambda}^{\lambda} \int d\hat{\mathbf{k}} \overline{Y_\alpha}(\hat{\mathbf{k}}) Y_{\lambda\mu}(\hat{\mathbf{k}}) Y_\beta(\hat{\mathbf{k}}) \overline{Y_{\lambda\mu}}(\hat{\mathbf{a}}) \\ &= \boxed{4\pi \sum_{\lambda=|l_\alpha-l_\beta|}^{l_\alpha+l_\beta} i^{l_\alpha-l_\beta-\lambda} \int_0^\infty dk k^2 \overline{I_\alpha}(k) I_\beta(k) j_\lambda(ka) \sum_{\mu=-\lambda}^{\lambda} \langle\alpha|\lambda\mu|\beta\rangle \overline{Y_{\lambda\mu}}(\hat{\mathbf{a}})} \quad (8.29) \end{aligned}$$

Eqn. (8.29) is the general overlap formula. $\langle\alpha|\lambda\mu|\beta\rangle$ is a Gaunt coefficient (or a linear combination of Gaunt coefficients, depending on the forms of the α and β orbitals). The index λ runs from $|l_\alpha - l_\beta|$ to $l_\alpha + l_\beta$ and jumps in steps of 2 (Eqn. (7.4)); The index μ runs from $-\lambda$ to λ , but only the μ 's which give non-trivial Gaunt coefficients contribute (Eqn. (7.3)).

To get a flavor of the overlap formula, consider the simplest case with two hydrogen $1s$ orbitals. In this case, $l_\alpha = l_\beta = 0$, the only non-trivial Gaunt coefficient is $\langle 00|00|00\rangle = 1/\sqrt{4\pi}$. Hence ($\lambda = \mu = 0$),

$$\langle 1s|1s(\mathbf{a})\rangle = \int_0^\infty dk k^2 |I_{1s}(k)|^2 j_0(ka) \quad (8.30)$$

Substituting the $1s$ wave function and the 0th spherical Bessel function,

$$R_{1s}(r) = 2e^{-r} \quad \text{and} \quad j_0(kr) = \frac{\sin(kr)}{kr} \quad (8.31)$$

8. Orbital overlaps and basis orthonormalization

$$I_{1s}(k) = \sqrt{\frac{2}{\pi}} \int_0^\infty dr r^2 j_0(kr) R_{1s}(r) = \sqrt{\frac{2}{\pi}} \frac{4}{(k^2 + 1)^2} \quad (8.32)$$

$$\langle 1s | 1s(\mathbf{a}) \rangle = \int_0^\infty dk k^2 |I_{1s}(k)|^2 j_0(ka) = \frac{1}{3}(a^2 + 3a + 3)e^{-a} \quad (8.33)$$

The overlap (8.33) describes exactly the picture shown in Fig. 8.1. If the two orbitals are both centered at the origin, the integral gives 1, which is the wave function normalization. If we pull the two orbitals apart, the overlap integral drops exponentially. Now we have understood the problem not only qualitatively, but also quantitatively.

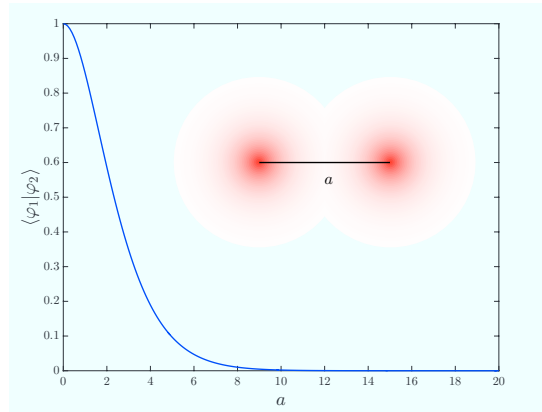


Figure 8.3.: Overlap integral between two hydrogen 1s orbitals separated at distance a (units in Bohr radius). When $a = 0$, the overlap is simply the wave function normalization. As a increases, the overlap drops exponentially.

Slater-Koster relations:

Eqn. (8.29) is an *explicit* formalism of the orbital overlaps in terms of atomic wave functions. Closely related are the Slater-Koster relations [55, 56].

Consider $\langle 2p_x | 2p_x(\mathbf{a}) \rangle$. The Slater-Koster relations provide the *decomposition*:

$$\langle 2p_x | 2p_x(\mathbf{a}) \rangle = c_1(pp\sigma) + c_2(pp\pi)$$

where,

$$(pp\sigma) = \langle 2p_x | 2p_x(a\hat{\mathbf{x}}) \rangle \quad \text{and} \quad (pp\pi) = \langle 2p_x | 2p_x(a\hat{\mathbf{y}}) \rangle = \langle 2p_x | 2p_x(a\hat{\mathbf{z}}) \rangle$$

The $(pp\sigma)$ and $(pp\pi)$ are treated as input parameters in the Slater-Koster relations. Now, from the overlap formula, we can work out the $(pp\sigma)$ and $(pp\pi)$. However, since the overlap formula is explicit, we have the advantage to skip the decomposition step and directly work out the overlap.

8.4. Orbital overlap angular dependencies

The overlap formula (8.29) can be written in a more compact form,

$$\langle \alpha | \beta(\mathbf{a}) \rangle = \sum_{\lambda=|l_\alpha-l_\beta|}^{l_\alpha+l_\beta} R_{\alpha\beta}^{(\lambda)}(a) A_{\alpha\beta}^{(\lambda)}(\hat{\mathbf{a}}) \quad (8.34)$$

with the radial component,

$$R_{\alpha\beta}^{(\lambda)}(a) \equiv \int_0^\infty dk \, k^2 \bar{I}_\alpha(k) I_\beta(k) j_\lambda(ka) \quad (8.35)$$

and the angular component,

$$A_{\alpha\beta}^{(\lambda)}(\hat{\mathbf{a}}) \equiv i^{l_\alpha-l_\beta-\lambda} 4\pi \sum_{\mu=-\lambda}^{\lambda} \langle \alpha | \lambda \mu | \beta \rangle \bar{Y}_{\lambda\mu}(\hat{\mathbf{a}}) \quad (8.36)$$

While the radial component is system specific, the angular component is universal. As an example, we work out the overlaps between the hydrogen orbitals: $2s$, $2p_x$, $2p_y$, and $2p_z$.

First, we work out the $I(k)$'s (8.24):

$$I_{2s}(k) = \sqrt{\frac{2}{\pi}} \int_0^\infty dr \, r^2 \left(\frac{\sin(kr)}{kr} \right) \left(\frac{1}{\sqrt{2}} \left(1 - \frac{1}{2}r \right) e^{-r/2} \right) = \frac{32(4k^2 - 1)}{\sqrt{\pi}(4k^2 + 1)^3} \quad (8.37)$$

$$I_{2p}(k) = \sqrt{\frac{2}{\pi}} \int_0^\infty dr \, r^2 \left(\frac{\sin(kr)}{(kr)^2} - \frac{\cos(kr)}{kr} \right) \left(\frac{1}{\sqrt{24}} r e^{-r/2} \right) = \frac{128k}{\sqrt{3\pi}(4k^2 + 1)^3} \quad (8.38)$$

Next, we calculate the radial component (8.35).

$$R_{2s,2s}^{(0)} = \frac{1}{240} (a^4 + 20a^2 + 120a + 240) e^{-a/2} \quad (8.39)$$

$$R_{2s,2p}^{(1)} = -\frac{\sqrt{3}}{720} (a^4 + 2a^3) e^{-a/2} \quad (8.40)$$

$$R_{2p,2p}^{(0)} = -\frac{1}{720} (a^4 - 60a^2 - 360a - 720) e^{-a/2} \quad (8.41)$$

$$R_{2p,2p}^{(2)} = \frac{1}{720} (a^4 + 6a^3 + 12a^2) e^{-a/2} \quad (8.42)$$

The angular component (8.36) can be obtained by inserting the Gaunt coefficients and the cubic harmonics. For instance,

$$\begin{aligned} A_{s,p_x}^{(1)} &= i^{-2} \frac{4\pi}{\sqrt{2}} \left(\langle 00 | 11 | 1, -1 \rangle \bar{Y}_{11}(\hat{\mathbf{a}}) - \langle 00 | 1, -1 | 11 \rangle \bar{Y}_{1,-1}(\hat{\mathbf{a}}) \right) \\ &= -\sqrt{2\pi} (Y_{1,-1}(\hat{\mathbf{a}}) - Y_{11}(\hat{\mathbf{a}})) = -\sqrt{3} \hat{a}_x \end{aligned} \quad (8.43)$$

where $\hat{a}_x = a_x/a$ (called the direction cosine) is the x -component of the unit vector $\hat{\mathbf{a}}$. All the angular dependences between s - and p - orbitals are given in Table 8.2.

8. Orbital overlaps and basis orthonormalization

Table 8.2.: Angular dependences of s - and p -orbital overlaps $A_{\alpha\beta}^{(\lambda)}(\hat{\mathbf{a}})$ (8.36). The unit vector $\hat{\mathbf{a}}$ points from the left indexed orbital to the right. $\hat{a}_x, \hat{a}_y, \hat{a}_z$ are the direction cosines. It might be helpful to summarize the results in more compact forms: $A_{s,s}^{(0)} = 1$, $A_{s,p_i}^{(1)} = -A_{p_i,s}^{(1)} = -\sqrt{3}\hat{a}_i$, $A_{p_i,p_j}^{(0)} = \delta_{ij}$, $A_{p_i,p_j}^{(2)} = \delta_{ij} - 3\hat{a}_i\hat{a}_j$. For table including d -orbitals, see Table A.6.

	s	p_x	p_y	p_z
s	$A^{(0)} = 1$	$A^{(1)} = -\sqrt{3}\hat{a}_x$	$A^{(1)} = -\sqrt{3}\hat{a}_y$	$A^{(1)} = -\sqrt{3}\hat{a}_z$
p_x	$A^{(1)} = \sqrt{3}\hat{a}_x$	$A^{(0)} = 1$ $A^{(2)} = 1 - 3\hat{a}_x^2$	$A^{(0)} = 0$ $A^{(2)} = -3\hat{a}_x\hat{a}_y$	$A^{(0)} = 0$ $A^{(2)} = -3\hat{a}_x\hat{a}_z$
p_y	$A^{(1)} = \sqrt{3}\hat{a}_y$	$A^{(0)} = 0$ $A^{(2)} = -3\hat{a}_x\hat{a}_y$	$A^{(0)} = 1$ $A^{(2)} = 1 - 3\hat{a}_y^2$	$A^{(0)} = 0$ $A^{(2)} = -3\hat{a}_y\hat{a}_z$
p_z	$A^{(1)} = \sqrt{3}\hat{a}_z$	$A^{(0)} = 0$ $A^{(2)} = -3\hat{a}_x\hat{a}_z$	$A^{(0)} = 0$ $A^{(2)} = -3\hat{a}_y\hat{a}_z$	$A^{(0)} = 1$ $A^{(2)} = 1 - 3\hat{a}_z^2$

Collecting the results, we visualize the overlaps between hydrogen $2s$, $2p_x$, $2p_y$, and $2p_z$ wave functions on the x - y plane (Fig. 8.5). Many geometric properties of the orbitals are reflected in the overlap pictures. For instance, consider $\langle 2s | 2p_x(\mathbf{a}) \rangle$ (Fig. 8.5b). If the $2p_x$ orbital is on the y -axis, the overlap is zero due to the symmetry (Fig. 8.4a), which is described by

$$A_{s,p_x}^{(\lambda)}(\hat{a}_x=0) = 0 \quad (8.44)$$

Another example, consider $\langle 2p_x | 2p_x(\mathbf{a}) \rangle$ along the x direction (Fig. 8.5f). Due to the oscillations of the two $2p_x$ wave functions, the overlap is positive if a_x is small (Fig. 8.4b), but negative if a_x is large (Fig. 8.4c).

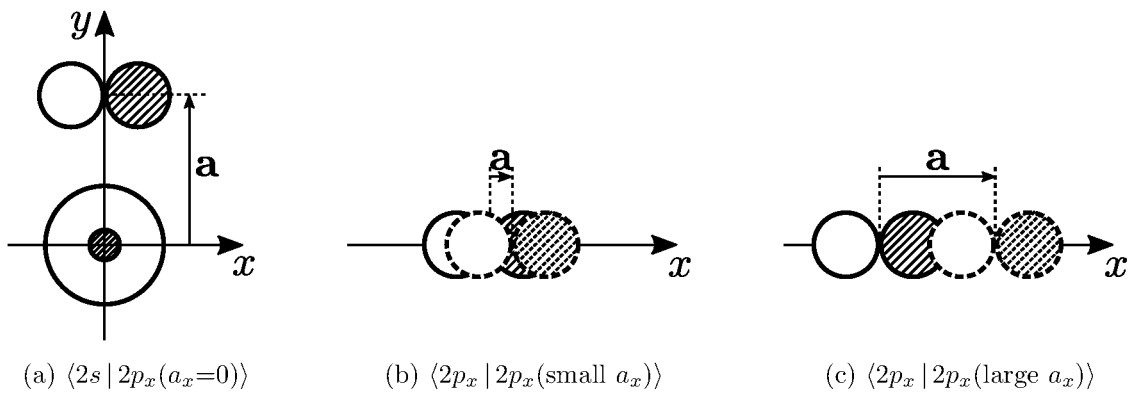


Figure 8.4.: (a) $2s$ - $2p_x$ on the y -axis (zero overlap due to symmetry); (b) $2p_x$ - $2p_x$ on the x -axis with small separation (positive overlap); (c) $2p_x$ - $2p_x$ on the x -axis with large separation (negative overlap).

8.4. Orbital overlap angular dependencies

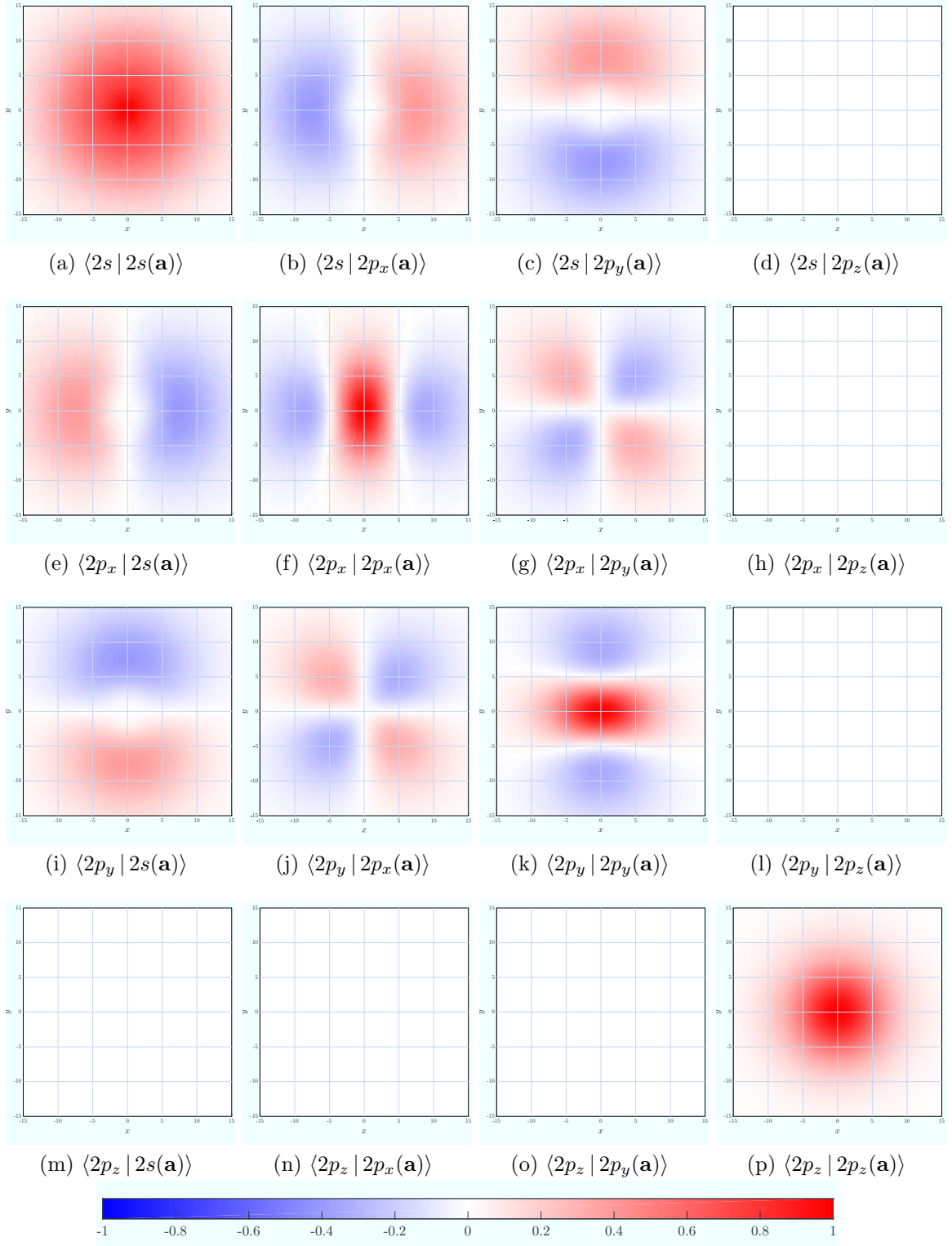


Figure 8.5.: Overlaps between hydrogen 2s, 2p_x, 2p_y, and 2p_z wave functions, visualized on the a_x - a_y plane ($a_z = 0$). The lengths are given in units of Bohr radius.

8.5. Basis orthonormalization

8.5.1. Generalized eigenvalue problem

To understand the eigenvalue problems under a non-orthogonal basis, we consider the simplest two-site problem: the hydrogen molecule ion H_2^+ , with the Hamiltonian,

$$H = -\frac{1}{2}\nabla^2 + V_1 + V_2 \quad (8.45)$$

where $V_1 = -\frac{1}{|\mathbf{r}|}$ is the potential centered at the origin and $V_2 = -\frac{1}{|\mathbf{r}-\mathbf{a}|}$ is the potential centered at location $\mathbf{a}$. The choice of basis set is the two $1s$ orbitals around the two nuclei:

$$|\varphi_1\rangle = |1s\rangle \quad \text{and} \quad |\varphi_2\rangle = |1s(\mathbf{a})\rangle$$

The matrix representation of the Hamiltonian is

$$\mathbf{H} = \begin{bmatrix} \langle\varphi_1|H|\varphi_1\rangle & \langle\varphi_1|H|\varphi_2\rangle \\ \langle\varphi_2|H|\varphi_1\rangle & \langle\varphi_2|H|\varphi_2\rangle \end{bmatrix} = \begin{bmatrix} X & Y \\ Y & X \end{bmatrix} \quad (8.46)$$

From (2.4) and (9.22),

$$X = \varepsilon_{1s} - \frac{1}{a} + \left(1 + \frac{1}{a}\right) e^{-2a} \quad (8.47)$$

From (2.5), (8.33), and (9.20),

$$Y = \frac{\varepsilon_{1s}}{3}(a^2 + 3a + 3)e^{-a} - (a + 1)e^{-a} \quad (8.48)$$

However, if we diagonalize the matrix $\mathbf{H}$ directly, we don't get the correct eigen-energies. The hidden problem is that the basis states $|\varphi_1\rangle$ and $|\varphi_2\rangle$ are *not* orthogonal. If we are working with non-orthogonal basis states, the corresponding matrix form of the eigenvalue problem is a *generalized eigenvalue problem*:

$$\mathbf{H}\mathbf{v} = E\mathbf{S}\mathbf{v} \quad (8.49)$$

where $\mathbf{S}$ is the overlap matrix

$$\mathbf{S} = \begin{bmatrix} \langle\varphi_1|\varphi_1\rangle & \langle\varphi_1|\varphi_2\rangle \\ \langle\varphi_2|\varphi_1\rangle & \langle\varphi_2|\varphi_2\rangle \end{bmatrix} = \begin{bmatrix} 1 & s \\ s & 1 \end{bmatrix} \quad (8.50)$$

The task is to solve the generalized eigenvalue problem:

$$\begin{bmatrix} X & Y \\ Y & X \end{bmatrix} \mathbf{v} = E \begin{bmatrix} 1 & s \\ s & 1 \end{bmatrix} \mathbf{v} \quad (8.51)$$

Solving the matrix problem, we obtain the eigen-energies:

$$E = \frac{X \pm Y}{1 \pm s} \quad (8.52)$$

Eqn. (8.52) is the solution considering the basis non-orthogonality, which converges to (2.7) and (2.8) if s is small.

8.5.2. Löwdin symmetric orthogonalization

Choosing a non-orthogonal basis leads to solving a generalized eigenvalue problem. In second quantization, working with a non-orthogonal basis can be extremely inconvenient since [57],

$$\{c_\alpha, c_\beta^\dagger\} = \langle \alpha | \beta \rangle \neq \delta_{\alpha\beta} \quad (8.53)$$

An alternative approach to handle a non-orthogonal basis is to perform a basis orthonormalization. We are looking for a transformation

$$|\perp_n\rangle = \sum_m |\varphi_m\rangle T_{mn} \quad \text{s.t.} \quad \langle \perp_m | \perp_n \rangle = \delta_{mn} \quad (8.54)$$

This implies,

$$\langle \perp_m | \perp_n \rangle = \sum_{ij} T_{ni}^\dagger \langle \varphi_i | \varphi_j \rangle T_{jm} = (T^\dagger S T)_{nm} = \delta_{nm}$$

equivalently,

$$\mathbf{S}^{-1} = \mathbf{T} \mathbf{T}^\dagger \quad (8.55)$$

However, Eqn. (8.55) does not determine the transformation matrix $\mathbf{T}$ uniquely. Indeed, there are various ways of performing basis orthonormalization, such as the Gram-Schmidt process or the Cholesky decomposition. Here we are interested in the Löwdin symmetric orthogonalization scheme, which has the advantage that the resulting orthonormal basis has the *minimum deformation* (see Fig: 8.7 and Fig: 8.8) from the original basis. Thus the original atomic features are kept as much as possible. The transformation matrix has a simple form [47]:

$$\boxed{\mathbf{T} = \mathbf{S}^{-\frac{1}{2}}} \quad (8.56)$$

If we perform the Löwdin symmetric orthogonalization scheme on the H_2^+ system, we obtain,

$$\mathbf{T} = \begin{bmatrix} 1 & s \\ s & 1 \end{bmatrix}^{-\frac{1}{2}} = \frac{1}{2} \begin{bmatrix} \frac{1}{\sqrt{1+s}} + \frac{1}{\sqrt{1-s}} & \frac{1}{\sqrt{1+s}} - \frac{1}{\sqrt{1-s}} \\ \frac{1}{\sqrt{1+s}} - \frac{1}{\sqrt{1-s}} & \frac{1}{\sqrt{1+s}} + \frac{1}{\sqrt{1-s}} \end{bmatrix} = \begin{bmatrix} c_1 & c_2 \\ c_2 & c_1 \end{bmatrix} \quad (8.57)$$

The two orthonormalized states are

$$|\perp_1\rangle = c_1 |\varphi_1\rangle + c_2 |\varphi_2\rangle \quad (8.58)$$

$$|\perp_2\rangle = c_2 |\varphi_1\rangle + c_1 |\varphi_2\rangle \quad (8.59)$$

Under the new basis, the Hamiltonian is,

$$\mathbf{H}^\perp = \begin{bmatrix} \langle \perp_1 | H | \perp_1 \rangle & \langle \perp_1 | H | \perp_2 \rangle \\ \langle \perp_2 | H | \perp_1 \rangle & \langle \perp_2 | H | \perp_2 \rangle \end{bmatrix} = \begin{bmatrix} A & B \\ B & A \end{bmatrix} \quad (8.60)$$

In fact,

$$\mathbf{H}^\perp = \mathbf{T}^\dagger \mathbf{H}^\varphi \mathbf{T} \quad (8.61)$$

Now,

$$\mathbf{H}^\perp = \begin{bmatrix} (c_1^2 + c_2^2)X + 2c_1c_2Y & (c_1^2 + c_2^2)Y + 2c_1c_2X \\ (c_1^2 + c_2^2)Y + 2c_1c_2X & (c_1^2 + c_2^2)X + 2c_1c_2Y \end{bmatrix} \quad (8.62)$$

8. Orbital overlaps and basis orthonormalization

Solving the *ordinary eigenvalue problem*:

$$\mathbf{H}^\perp \mathbf{v} = E \mathbf{v} \quad (8.63)$$

yields eigen-energies:

$$E = A \pm B = (c_1 \pm c_2)^2 (X \pm Y) = \frac{X \pm Y}{1 \pm s} \quad (8.64)$$

which agree with the results in (8.52) from the generalized eigenvalue problem. Notice that the basis orthonormalization is *not* a unitary transformation, as obviously the inner products are not preserved. The necessity of the basis orthonormalization can be understood from the following: Fig. 8.6 plots the eigen-energies of the hydrogen molecule ion system (plus the proton-proton potential energy $1/a$) obtained by (Fig. 8.6a) diagonalizing the matrix representation $\mathbf{H}^\varphi$ under the original non-orthogonal atomic basis without the overlap matrix; (Fig. 8.6b) diagonalizing the matrix representation $\mathbf{H}^\perp$ under the orthogonalized basis. The bond length of the system is properly described under the orthogonalized basis.

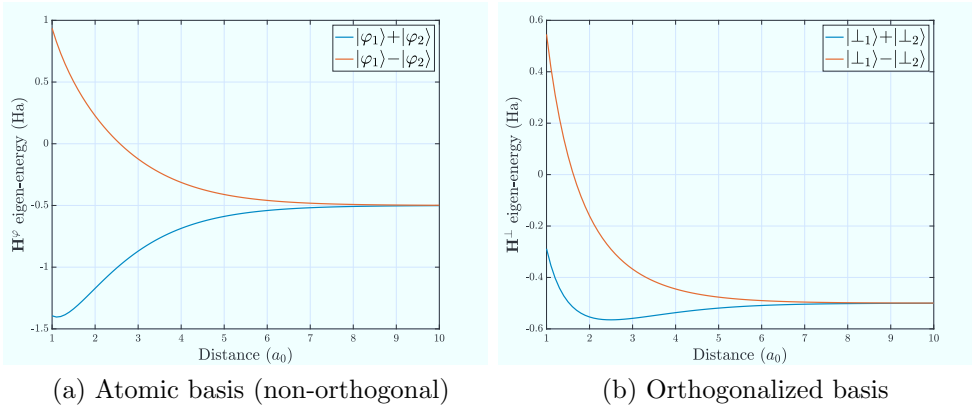


Figure 8.6.: Eigen-energies of the hydrogen molecule ion system obtained from (a) diagonalizing the matrix representation $\mathbf{H}^\varphi$ under the original non-orthogonal atomic basis without the overlap matrix; (b) diagonalizing the matrix representation $\mathbf{H}^\perp$ under the orthogonalized basis. The bond length of the system is properly described under the orthogonalized basis.

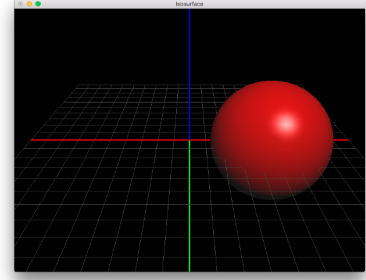
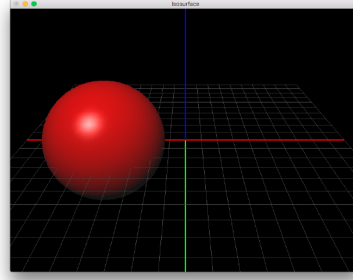
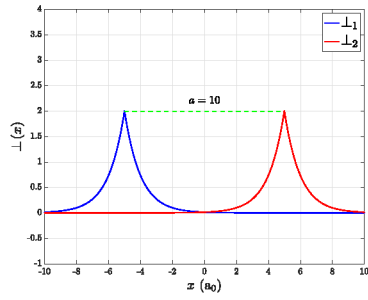
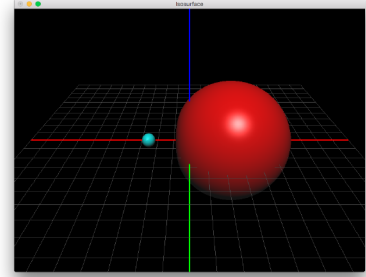
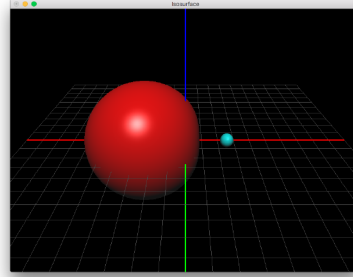
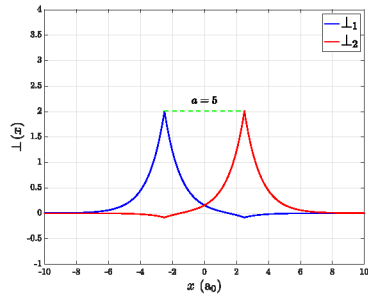
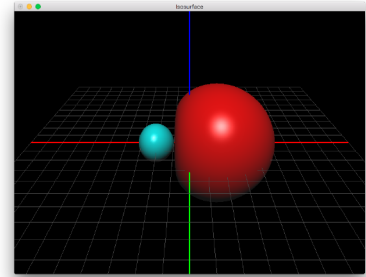
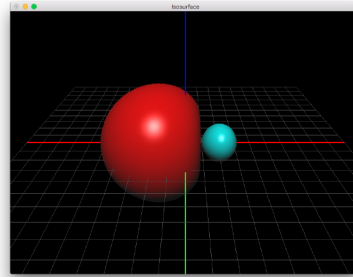
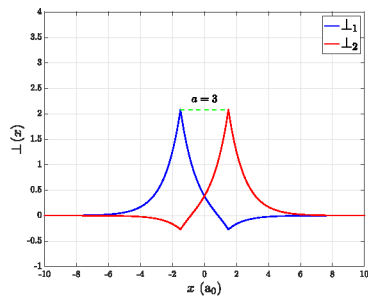
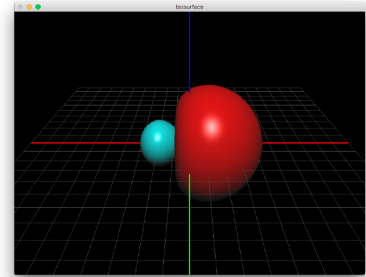
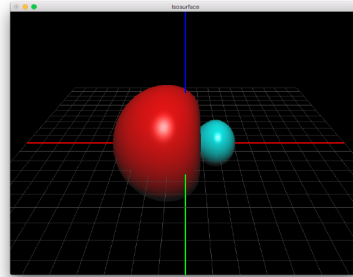
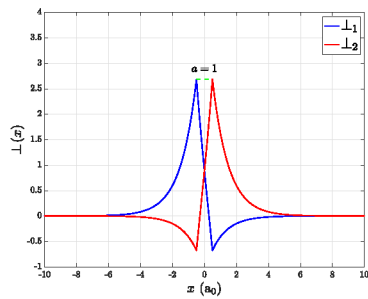
(a) $a = 10 a_0$ $s^\varphi = 0.002$ $s^\perp = 0$ (b) $a = 5 a_0$ $s^\varphi = 0.097$ $s^\perp = 0$ (c) $a = 3 a_0$ $s^\varphi = 0.349$ $s^\perp = 0$ (d) $a = 1 a_0$ $s^\varphi = 0.858$ $s^\perp = 0$

Figure 8.7.: Deformation of two hydrogen 1s orbitals under the Löwdin transformations. First column: plots of the two orthonormalized radial wave functions; Right two columns: isosurface plots of the two orthonormalized wave functions. The isosurfaces are generated by a visualization program developed by Khaldoun Ghanem during the JSC Guest Student Programme 2012.

8. Orbital overlaps and basis orthonormalization

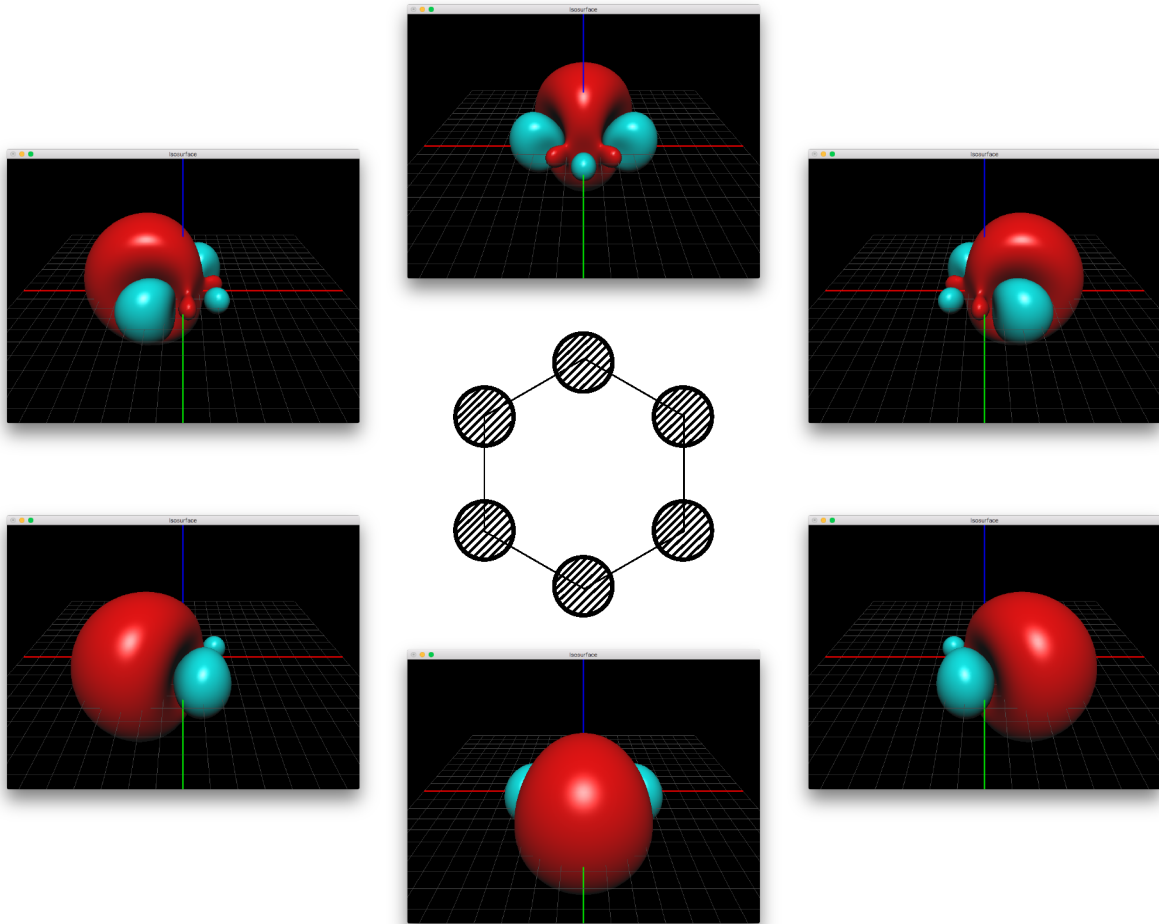


Figure 8.8.: Isosurface plots of the orthonormalized orbitals of a six-site $1s$ ring-system with neighboring distance $a = 2a_0$.

9. Re-centering method and multi-center matrix elements

9.1. Re-centering method

In the previous chapter, we performed the overlap integrals by transforming the two atomic wave functions into plane waves. That was perhaps the simplest approach to understand and to solve the overlap problems. In this chapter, we consider more general integrals that involve multi-center spherical functions. It is more convenient to view the problems from a re-centering perspective. We solve a multi-center integral by the following steps,

1. Expand each off-centered spherical functions into off-centered spherical waves;
2. Expand each off-centered spherical waves into centered spherical waves;
3. Integrate over the centered spherical waves.

Apparently, the second step is the key where the “re-centering” takes place. The task is to expand an off-centered spherical wave $|k\alpha(\mathbf{a})\rangle$ in terms of (a lot of) centered spherical waves $|k'\alpha'\rangle$:

$$\begin{aligned}
 |k\alpha(\mathbf{a})\rangle &= \int_0^\infty dk' \sum_{\alpha'} |k'\alpha'\rangle \langle k'\alpha' | k\alpha(\mathbf{a})\rangle \\
 &= \int_0^\infty dk' \sum_{\alpha'} |k'\alpha'\rangle \langle k'\alpha' | \left(\int d\mathbf{k}'' |\mathbf{k}''\rangle \langle \mathbf{k}''| \right) |k\alpha(\mathbf{a})\rangle \\
 &= \int_0^\infty dk' \sum_{\alpha'} \int d\mathbf{k}'' |k'\alpha'\rangle \langle k'\alpha' | \mathbf{k}''\rangle \langle \mathbf{k}'' | k\alpha(\mathbf{a})\rangle
 \end{aligned} \tag{9.1}$$

Recall (8.11), we have,

$$\langle k'\alpha' | \mathbf{k}''\rangle = \frac{i^{l_{\alpha'}}}{k'} \overline{Y_{\alpha'}(\hat{\mathbf{k}}'')} \delta(k' - k'') \tag{9.2}$$

$$\langle \mathbf{k}'' | k\alpha(\mathbf{a})\rangle = \frac{i^{-l_\alpha}}{k} Y_\alpha(\hat{\mathbf{k}}'') \delta(k - k'') e^{-i\mathbf{k}'' \cdot \mathbf{a}} \tag{9.3}$$

9. Re-centering method and multi-center matrix elements

Thus,

$$\begin{aligned}
|k\alpha(\mathbf{a})\rangle &= \sum_{\alpha'} i^{l_{\alpha'}-l_{\alpha}} \int d\hat{\mathbf{k}} \overline{Y_{\alpha'}(\hat{\mathbf{k}})} Y_{\alpha}(\hat{\mathbf{k}}) e^{-i\mathbf{k}\cdot\mathbf{a}} |k\alpha'\rangle \\
&= \sum_{\alpha'} i^{l_{\alpha'}-l_{\alpha}} \int d\hat{\mathbf{k}} \overline{Y_{\alpha'}(\hat{\mathbf{k}})} Y_{\alpha}(\hat{\mathbf{k}}) \left[4\pi \sum_{\lambda=0}^{\infty} i^{-\lambda} j_{\lambda}(ka) \sum_{\mu=-\lambda}^{\lambda} Y_{\lambda\mu}(\hat{\mathbf{k}}) \overline{Y_{\lambda\mu}(\hat{\mathbf{a}})} \right] |k\alpha'\rangle \\
&= 4\pi \sum_{\alpha'} \sum_{\lambda\mu} i^{l_{\alpha'}-l_{\alpha}-\lambda} \langle\alpha'|\lambda\mu|\alpha\rangle j_{\lambda}(ka) \overline{Y_{\lambda\mu}(\hat{\mathbf{a}})} |k\alpha'\rangle
\end{aligned} \tag{9.4}$$

In a compact notation, we write down the re-centering formula:

$$\boxed{|k\alpha(\mathbf{a})\rangle = \sum_{\alpha'} g_{\alpha'\alpha}(k\mathbf{a}) |k\alpha'\rangle} \tag{9.5}$$

where,

$$\boxed{g_{\alpha'\alpha}(k\mathbf{a}) \equiv 4\pi \sum_{\lambda\mu} i^{l_{\alpha'}-l_{\alpha}-\lambda} \langle\alpha'|\lambda\mu|\alpha\rangle j_{\lambda}(ka) \overline{Y_{\lambda\mu}(\hat{\mathbf{a}})}} \tag{9.6}$$

The indices λ and μ run for non-trivial Gaunt coefficients (7.4) and (7.3).

It is quite interesting to think on the re-centering formula (9.5): keep adding spherical waves centered at the origin, you eventually recover a spherical wave centered at location $\mathbf{a}$. How is that possible? To understand how the re-centering formula works, we consider a simple $l = 0$ spherical wave with a displacement on the z -axis ($\mathbf{a} = a\hat{\mathbf{z}}$).

$$|ks(a\hat{\mathbf{z}})\rangle = \sum_{l'm'} g_{l'm',s}(ka\hat{\mathbf{z}}) |kl'm'\rangle \tag{9.7}$$

where,

$$g_{l'm',s}(ka\hat{\mathbf{z}}) = 4\pi \underbrace{\langle l'm'|l'm'|s\rangle}_{\frac{1}{\sqrt{4\pi}}} j_{l'}(ka) \underbrace{\overline{Y_{l'm'}(\hat{\mathbf{z}})}}_{\sqrt{\frac{2l'+1}{4\pi}} \delta_{m'0}} = \sqrt{2l'+1} j_{l'}(ka) \delta_{m'0} \tag{9.8}$$

Hence,

$$|ks(a\hat{\mathbf{z}})\rangle = \sum_{l'=0}^{\infty} \sqrt{2l'+1} j_{l'}(ka) |kl'0\rangle \tag{9.9}$$

To construct a shifted $|ks(a\hat{\mathbf{z}})\rangle$, we add up many centered $|kl'0\rangle$ with the corresponding weights. But, to a good approximation, how many centered waves do we need to recover the shifted wave? Fig. 9.1 visualizes the results from adding up 5, 10, 15, and 20 waves. With the increasing number of centered waves, the sum recovers a clearer shifted wave.

To understand the cut-off problem quantitatively, we sum up l' from 0 to l_{cut} and check

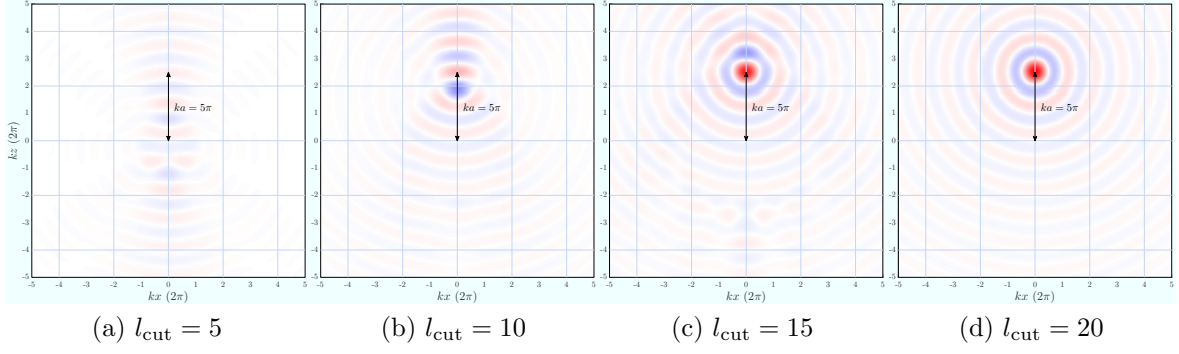


Figure 9.1.: Recovering an off-centered spherical wave, by accumulating centered spherical waves.

the overlap between $|ks(a\hat{\mathbf{z}})\rangle_{\text{cut}}$ and the exact $|ks(a\hat{\mathbf{z}})\rangle$.

$$\begin{aligned} {}_{\text{cut}}\langle ks(a\hat{\mathbf{z}}) | ks(a\hat{\mathbf{z}}) \rangle &= \sum_{l'=0}^{l_{\text{cut}}} \sum_{l''=0}^{\infty} \sqrt{(2l'+1)(2l''+1)} j_{l'}(ka) j_{l''}(ka) \underbrace{\langle kl'0 | kl''0 \rangle}_{\delta(0) \delta_{l'l''}} \\ &= \delta(0) \sum_{l'=0}^{l_{\text{cut}}} (2l'+1) j_{l'}^2(ka) \end{aligned} \quad (9.10)$$

The problem reduces to finding l_{cut} such that

$$1 - \sum_{l'=0}^{l_{\text{cut}}} (2l'+1) j_{l'}^2(ka) \leq \text{tol} \quad (9.11)$$

This well defines the number of centered spherical waves required to reconstruct a decent shifted wave. It should be rather intuitive that the further the off-centering is, the larger l_{cut} is required. With $\text{tol} = 10^{-8}$, Fig. 9.2 plots the l_{cut} required against displacement, which shows a nice linear behavior. For a displacement of $ka = 5\pi$, we need $l_{\text{cut}} = 25$.

In general, we consider re-centering an arbitrary spherical wave with displacement in the $\hat{\mathbf{z}}$ direction.

$$\begin{aligned} |klm(a\hat{\mathbf{z}})\rangle &= \sum_{l'm'} g_{l'm',lm}(ka\hat{\mathbf{z}}) |kl'm'\rangle \\ &= \sqrt{4\pi} \sum_{l'=0}^{\infty} \sum_{\lambda} i^{l'-l-\lambda} j_{\lambda}(ka) \sqrt{2\lambda+1} \langle l'm | \lambda 0 | lm \rangle |kl'm\rangle \end{aligned} \quad (9.12)$$

9. Re-centering method and multi-center matrix elements

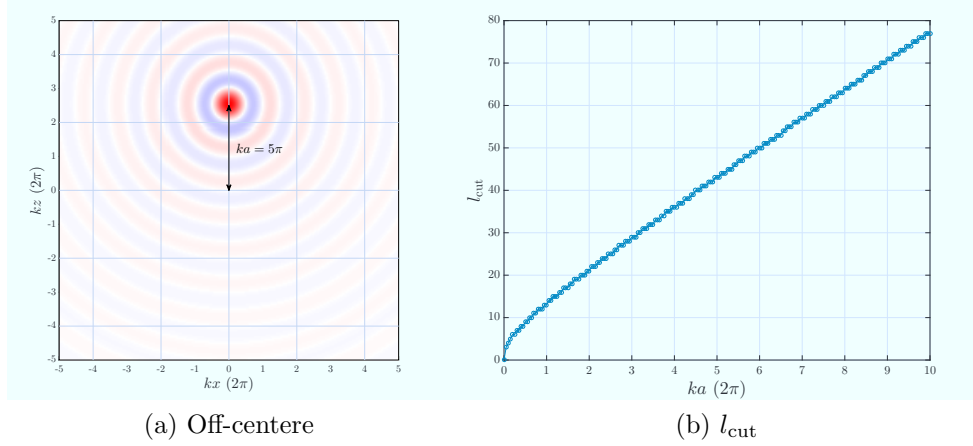


Figure 9.2.: (a) Off-centered s spherical wave with displacement $ka = 5\pi$ in the $\hat{z}$ direction, reconstructed by centered spherical waves with $l_{\text{cut}} = 25$. (b) l_{cut} for decent wave recentering ($\text{tol} = 10^{-8}$).

Now, we consider the overlap with cut-off:

$$\begin{aligned}
 {}_{\text{cut}}\langle klm(a\hat{z}) | klm(a\hat{z}) \rangle &= 4\pi \sum_{l'=0}^{l_{\text{cut}}} \sum_{l''=0}^{\infty} \sum_{\lambda'\lambda''} i^{-l'+\lambda'+l''-\lambda''} j_{\lambda'}(ka) j_{\lambda''}(ka) \\
 &\quad \sqrt{(2\lambda'+1)(2\lambda''+1)} \langle l'm | \lambda'0 | lm \rangle \langle l''m | \lambda''0 | lm \rangle \langle kl'm | kl''m \rangle \\
 &= \delta(0) 4\pi \sum_{l'=0}^{l_{\text{cut}}} \sum_{\lambda'\lambda''} i^{\lambda'-\lambda''} j_{\lambda'}(ka) j_{\lambda''}(ka) \\
 &\quad \sqrt{(2\lambda'+1)(2\lambda''+1)} \langle l'm | \lambda'0 | lm \rangle \langle l'm | \lambda''0 | lm \rangle
 \end{aligned} \tag{9.13}$$

The problem reduces to finding l_{cut} such that

$$1 - 4\pi \sum_{l'=0}^{l_{\text{cut}}} \sum_{\lambda'\lambda''} i^{\lambda'-\lambda''} j_{\lambda'}(ka) j_{\lambda''}(ka) \sqrt{(2\lambda'+1)(2\lambda''+1)} \langle l'm | \lambda'0 | lm \rangle \langle l'm | \lambda''0 | lm \rangle \leq \text{tol} \tag{9.14}$$

Condition (9.14) is a generalization of (9.11) for re-centering arbitrary spherical waves. For different l 's and m 's, l_{cut} are plotted in Fig. 9.3 with $\text{tol} = 10^{-8}$. We discovered that the l_{cut} for spherical waves with quantum number l, m is almost the same as the s spherical wave but with a constant shift l (there is a weak dependence on m , but $m = 0$ always give a safer estimation):

$$l_{\text{cut}}^{(l,m)} = l_{\text{cut}}^{(0)} + l \tag{9.15}$$

where $l_{\text{cut}}^{(0)}$ is given by Eqn. (9.11).

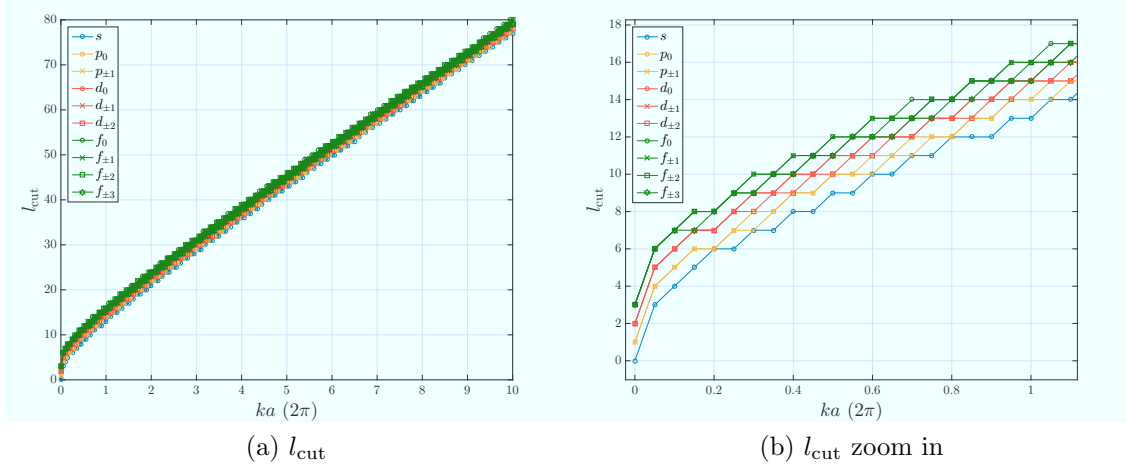


Figure 9.3.: (a) l_{cut} for decent s , p , d , and f wave recentering (tol = 10^{-8}). (b) Zoom in plot shows the constant l -shift originates from the beginning.

9.2. Overlap problem revisited

We can derive the overlap formula (8.29) from the re-centering formula (9.5) in a compact way. Here we reproduce the overlap integral from the re-centering perspective instead of the plane wave expansion.

First, we expand the off-centered atomic orbital in terms of off-centered spherical waves (8.25):

$$|\beta(\mathbf{a})\rangle = \int_0^\infty dk \, k I_\beta(k) |k\beta(\mathbf{a})\rangle \quad (9.16)$$

Next, we re-center the spherical waves (9.5):

$$|\beta(\mathbf{a})\rangle = \int_0^\infty dk \, k I_\beta(k) \sum_{\beta'} g_{\beta'\beta}(k\mathbf{a}) |k\beta'\rangle \quad (9.17)$$

The overlap reads,

$$\begin{aligned} \langle\alpha|\beta(\mathbf{a})\rangle &= \int_0^\infty dk \, k I_\beta(k) \sum_{\beta'} g_{\beta'\beta}(k\mathbf{a}) \underbrace{\langle\alpha|k\beta'\rangle}_{k\overline{I}_\alpha(k)\delta_{\alpha\beta'}} \\ &= \boxed{\int_0^\infty dk \, k^2 \overline{I}_\alpha(k) I_\beta(k) g_{\alpha\beta}(k\mathbf{a})} \end{aligned} \quad (9.18)$$

which recovers Eqn. (8.29) in a very compact way.

9.3. Hopping matrix elements

In general, a hopping matrix element is a three-center integral,

$$\langle \alpha(\mathbf{a}) | V | \beta(\mathbf{b}) \rangle$$

Here, we consider three simplified versions of the hopping elements:

$$\begin{aligned} h_{\alpha\beta}^{(1)}(\mathbf{a}) &= \langle \alpha | \frac{1}{r} | \beta(\mathbf{a}) \rangle \\ h_{\alpha\beta}^{(2)}(\mathbf{a}) &= \langle \alpha(\mathbf{a}) | \frac{1}{r} | \beta(\mathbf{a}) \rangle \\ h_{s_1 s_2}(\mathbf{a}, \mathbf{b}) &= \langle s_1(\mathbf{a}) | \frac{1}{r} | s_2(\mathbf{b}) \rangle \end{aligned}$$

Element $h_{\alpha\beta}^{(1)}$ appeared in (2.5). In this case, orbital- α and the potential $\frac{1}{r}$ are centered at the same site. This two-center integral is actually identical to the overlap integral up to a rescaling,

$$\boxed{h_{\alpha\beta}^{(1)}(\mathbf{a}) = \langle \tilde{\alpha} | \beta(\mathbf{a}) \rangle} \quad \text{where} \quad \tilde{\varphi}_{\alpha}(\mathbf{r}) = \frac{\varphi_{\alpha}(\mathbf{r})}{r} \quad (9.19)$$

Consider the hydrogen $1s$ orbital ($\alpha = \beta = H_{1s}$). $h_{1s}^{(1)}$ is analogous to (8.33),

$$h_{1s}^{(1)}(\mathbf{a}) = \left\langle \tilde{1s} \left| 1s(\mathbf{a}) \right. \right\rangle = \int_0^{\infty} dk \, k^2 \bar{I}_{1s}(k) I_{1s}(k) j_0(ka) = (a+1)e^{-a} \quad (9.20)$$

Element $h_{\alpha\beta}^{(2)}$ appeared in (2.4). In this case, orbitals α and β are on one site but the potential is centered on another site. We solve this integral by translating the coordinates and using the multipole expansion (4.11),

$$\begin{aligned} h_{\alpha\beta}^{(2)}(\mathbf{a}) &= \langle \alpha | \frac{1}{|\mathbf{r} + \mathbf{a}|} | \beta \rangle \\ &= \langle \alpha | \sum_{\lambda\mu} \frac{4\pi}{2\lambda+1} \frac{r_{\leq}^{\lambda}}{r_{>}^{\lambda+1}} Y_{\lambda\mu}(\hat{\mathbf{r}}) \overline{Y_{\lambda\mu}}(-\hat{\mathbf{a}}) | \beta \rangle \\ &= \boxed{\sum_{\lambda\mu} \frac{4\pi}{2\lambda+1} \langle R_{\alpha} | \frac{r_{\leq}^{\lambda}}{r_{>}^{\lambda+1}} | R_{\beta} \rangle \langle Y_{\alpha} | Y_{\lambda\mu} | Y_{\beta} \rangle \overline{Y_{\lambda\mu}}(-\hat{\mathbf{a}})} \end{aligned} \quad (9.21)$$

Consider $\alpha = \beta = H_{1s}$. The sum indices collapse to $\lambda = \mu = 0$,

$$h_{1s}^{(2)}(\mathbf{a}) = \langle R_{1s} | \frac{1}{r_{>}} | R_{1s} \rangle = \int_0^a dr \, \frac{1}{a} |u_{1s}|^2 + \int_a^{\infty} dr \, \frac{1}{r} |u_{1s}|^2 = \frac{1}{a} - \left(1 + \frac{1}{a}\right) e^{-2a} \quad (9.22)$$

Note that $h_{\alpha\beta}^{(1)}$ and $h_{\alpha\beta}^{(2)}$ elements have different decaying behaviors: $h_{\alpha\beta}^{(1)}$ decays exponentially with the separation due to the overlap decay; but $h_{\alpha\beta}^{(2)}$ drops proportional to the distance inverse which is the decaying of the off-site potential.

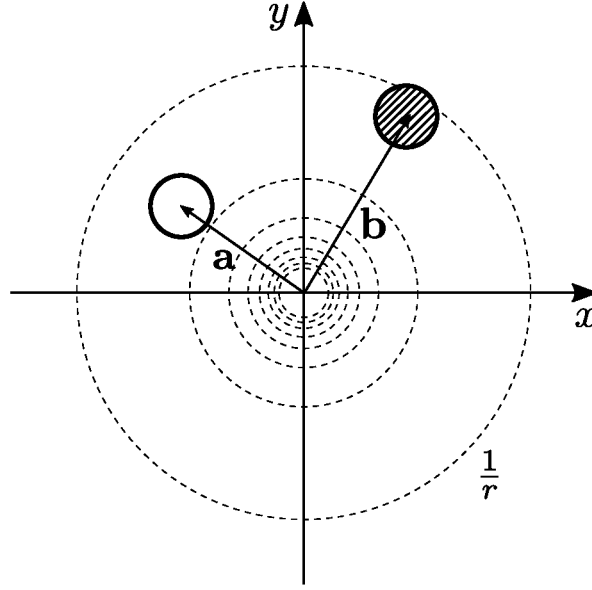


Figure 9.4.: Configuration of the matrix element $h_{s_1 s_2}(\mathbf{a}, \mathbf{b}) = \langle s_1(\mathbf{a}) | \frac{1}{r} | s_2(\mathbf{b}) \rangle$.

Element $h_{s_1 s_2}$ is a general three-center integral for spherically symmetric orbitals, namely, the s -orbitals. The physical picture of the matrix element is shown in Fig. 9.4. To solve this general three-center integral, we use the re-centering formula to bring the off-centered orbitals back to the origin:

$$h_{s_1 s_2}(\mathbf{a}, \mathbf{b}) = \int dk \int dk' k k' \overline{I_{s_1}}(k) I_{s_2}(k') \sum_{lm, l'm'} \overline{g_{lm, s}}(k\mathbf{a}) g_{l'm', s}(k'\mathbf{b}) \langle klm | \frac{1}{r} | k'l'm' \rangle \quad (9.23)$$

where (from (8.24) and (9.6)),

$$I_\alpha(k) = \sqrt{\frac{2}{\pi}} \int_0^\infty dr r^2 j_0(kr) R_\alpha(r) \quad (9.24)$$

$$g_{lm, s}(k\mathbf{a}) = 4\pi \langle lm | lm | s \rangle j_l(ka) \overline{Y_{lm}}(\hat{\mathbf{a}}) = \sqrt{4\pi} j_l(ka) \overline{Y_{lm}}(\hat{\mathbf{a}}) \quad (9.25)$$

and the “centered” integral,

$$\langle klm | \frac{1}{r} | k'l'm' \rangle = \langle R_{kl} | \frac{1}{r} | R_{k'l'} \rangle \langle Y_{lm} | Y_{l'm'} \rangle = \langle R_{kl} | \frac{1}{r} | R_{k'l'} \rangle \delta_{ll'} \delta_{mm'} \quad (9.26)$$

where $R_{kl}(r) = \sqrt{\frac{2}{\pi}} k j_l(kr)$ is the radial part of the spherical wave. Collecting the results,

9. Re-centering method and multi-center matrix elements

(9.23) becomes,

$$\begin{aligned}
 h_{s_1 s_2}(\mathbf{a}, \mathbf{b}) &= \int dk \int dk' k k' \overline{I_{s_1}(k)} I_{s_2}(k') \sum_l j_l(ka) j_l(k'b) \langle R_{kl} | \underbrace{\frac{1}{r} | R_{k'l} \rangle 4\pi \sum_m Y_{lm}(\hat{\mathbf{a}}) \overline{Y_{lm}}(\hat{\mathbf{b}})}_{(2l+1)P_l(\hat{\mathbf{a}} \cdot \hat{\mathbf{b}})} \\
 &= \sum_l (2l+1) \int dk \int dk' k k' \overline{I_{s_1}(k)} I_{s_2}(k') j_l(ka) j_l(k'b) \langle R_{kl} | \frac{1}{r} | R_{k'l} \rangle P_l(\hat{\mathbf{a}} \cdot \hat{\mathbf{b}}) \\
 &= \boxed{\sum_{l=0}^{\infty} (2l+1) \langle J_{s_1}^{(l)}(r, a) | \frac{1}{r} | J_{s_2}^{(l)}(r, b) \rangle P_l(\hat{\mathbf{a}} \cdot \hat{\mathbf{b}})} \quad (9.27)
 \end{aligned}$$

where we define the transformed radial function,

$$\boxed{J_{\alpha}^{(l)}(r, a) \equiv \sqrt{\frac{2}{\pi}} \int_0^{\infty} dk k^2 I_{\alpha}(k) j_l(kr) j_l(ka)} \quad (9.28)$$

Evaluating the transformed functions $J_{\alpha}^{(l)}$ is numerically straightforward (1-dimensional integral over k), but analytically challenging, as the integrals involve combinations of general spherical Bessel functions and radial wave functions with various forms. Here we use the hydrogen $1s$ -orbital as an example, and try to work out the $J_{1s}^{(l)}$ analytically.

We have worked out $I_{1s}(k)$ in Eqn. (8.32). Now, our task is to evaluate the integral of the form:

$$J_{1s}^{(l)}(r_1, r_2) = \frac{8}{\pi} \int_0^{\infty} dk \frac{k^2}{(k^2 + 1)^2} j_l(r_1 k) j_l(r_2 k) \quad (9.29)$$

To work out (9.29), we use the contour integration method. Note the integrand has poles (of order 2) at $\pm i$.

$$\frac{k^2}{(k^2 + 1)^2} = \frac{k^2}{(k + i)^2(k - i)^2} \quad (9.30)$$

We first realize that the integrand is symmetric about k (DLMF 10.47.14) [54]. This helps us to make the contour integral possible.

$$\int_0^{\infty} dk = \frac{1}{2} \int_{-\infty}^{+\infty} dk$$

Notice that r_1 and r_2 are exchangeably symmetric $J_{1s}^{(l)}(r_1, r_2) = J_{1s}^{(l)}(r_2, r_1) = J_{1s}^{(l)}(r_>, r_<)$, where $r_> = \max(r_1, r_2)$ and $r_< = \min(r_1, r_2)$. We can write,

$$J_{1s}^{(l)}(r_1, r_2) = \frac{4}{\pi} \int_{-\infty}^{+\infty} dk \frac{k^2}{(k + i)^2(k - i)^2} j_l(r_> k) j_l(r_< k) \quad (9.31)$$

To perform the real-axis integration using the “contour trick”, we must let the integrand vanish at infinity on the imaginary-axis. We achieve this by expressing the $j_l(r_> k)$ term

9.3. Hopping matrix elements

as $\frac{1}{2}[h_l^{(1)}(r_{>k}) + h_l^{(2)}(r_{>k})]$ (DLMF 10.47.10), where $h_l^{(1)}$ and $h_l^{(2)}$ are the *spherical Hankel functions* of the first and second kinds, respectively [54].

$$J_{1s}^{(l)}(r_1, r_2) = \frac{2}{\pi} \int_{-\infty}^{+\infty} dk \frac{k^2}{(k+i)^2(k-i)^2} \left[h_l^{(1)}(r_{>k}) j_l(r_{<k}) + h_l^{(2)}(r_{>k}) j_l(r_{<k}) \right] \quad (9.32)$$

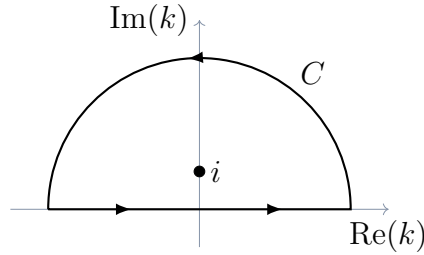
Using the reflection formulas $j_l(-z) = (-1)^l j_l(z)$ (DLMF 10.47.14) and $h_l^{(1)}(-z) = (-1)^l h_l^{(2)}(z)$ (DLMF 10.47.15) [54], we can combine the two terms.

$$J_{1s}^{(l)}(r_1, r_2) = \frac{4}{\pi} \int_{-\infty}^{+\infty} dk \frac{k^2}{(k+i)^2(k-i)^2} h_l^{(1)}(r_{>k}) j_l(r_{<k}) \quad (9.33)$$

We are ready to perform the contour integral as the integrand has the proper asymptotic behavior for $k \rightarrow \infty$ (DLMF 10.52.4) [54],

$$h_l^{(1)}(r_{>k}) j_l(r_{<k}) \sim e^{i(r_{>}+r_{<})k} + e^{i(r_{>} - r_{<})k} \quad (9.34)$$

As a consequence, we can take an infinitely large upper semi-circle as our contour, where the contribution on the upper arc vanishes. Therefore, we should solve the following contour integral problem.



$$J_{1s}^{(l)}(r_1, r_2) = \frac{4}{\pi} \oint_C dk \frac{k^2}{(k+i)^2(k-i)^2} h_l^{(1)}(r_{>k}) j_l(r_{<k}) \quad (9.35)$$

Inside the contour C , the integrand has one pole (of order 2) at $k = i$. The residual is,

$$\begin{aligned} \text{Res}_{k=i} &= \frac{d}{dk} \left[\frac{k^2}{(k+i)^2} h_l^{(1)}(r_{>k}) j_l(r_{<k}) \right]_{k=i} \\ &= -\frac{i}{4} h_l^{(1)}(r_{>i}) j_l(r_{<i}) + \frac{r_{>}}{4} h_l^{(1)'}(r_{>i}) j_l(r_{<i}) + \frac{r_{<}}{4} h_l^{(1)}(r_{>i}) j_l'(r_{<i}) \end{aligned} \quad (9.36)$$

To get rid of the first-order derivative terms, we employ the recursion relation $f_l'(z) = \frac{l}{z} f_l(z) - f_{l+1}(z)$ (DLMF 10.51.2) [54],

$$\begin{aligned} \text{Res}_{k=i} &= -\frac{i}{4} h_l^{(1)}(r_{>i}) j_l(r_{<i}) + \frac{r_{>}}{4} \left[\frac{l}{r_{>i}} h_l^{(1)}(r_{>i}) - h_{l+1}^{(1)}(r_{>i}) \right] j_l(r_{<i}) + \frac{r_{<}}{4} h_l^{(1)}(r_{>i}) \left[\frac{l}{r_{<i}} j_l(r_{<i}) - j_{l+1}(r_{<i}) \right] \\ &= -\frac{(2l+1)i}{4} h_l^{(1)}(r_{>i}) j_l(r_{<i}) - \frac{r_{>}}{4} h_{l+1}^{(1)}(r_{>i}) j_l(r_{<i}) - \frac{r_{<}}{4} h_l^{(1)}(r_{>i}) j_{l+1}(r_{<i}) \end{aligned} \quad (9.37)$$

9. Re-centering method and multi-center matrix elements

We can transform the imaginary arguments to real by

$$\begin{cases} j_l(iz) = i^l i_l^{(1)}(z) & (\text{DLMF 10.47.12}) \\ h_l^{(1)}(iz) = -\frac{2}{\pi} i^{-l} k_l(z) & (\text{DLMF 10.47.13}) \end{cases}$$

where $i_l^{(1)}(z)$ and $k_l(z)$ are the *modified spherical Bessel functions* [54].

The contour integral is given by the residue theorem [58]:

$$\begin{aligned} (2\pi i) \text{Res}_{k=i} &= -(2l+1)k_l(r_>)i_l^{(1)}(r_<) + r_>k_{l+1}(r_>)i_l^{(1)}(r_<) - r_<k_l(r_>)i_{l+1}^{(1)}(r_<) \\ &= -\underbrace{[(2l+1)k_l(r_>) - r_>k_{l+1}(r_>)]}_{r_>k_{l-1}(r_>) \text{ (DLMF 10.51.4)}} i_l^{(1)}(r_<) - r_<k_l(r_>)i_{l+1}^{(1)}(r_<) \\ &= r_>k_{l-1}(r_>)i_l^{(1)}(r_<) - r_<k_l(r_>)i_{l+1}^{(1)}(r_<) \end{aligned} \quad (9.38)$$

Finally,

$$J_{1s}^{(l)}(r_1, r_2) = \frac{4}{\pi} \left[r_>k_{l-1}(r_>)i_l^{(1)}(r_<) - r_<k_l(r_>)i_{l+1}^{(1)}(r_<) \right] \quad (9.39)$$

Algorithm 9.1 provides an example routine for computing $h_{1s}(\mathbf{a}, \mathbf{b})$ (9.27). To be consistent with our earlier discussions, we perform the radial integrations numerically on the logarithmic grid (3.12).

Algorithm 9.1: Routines for computing $h_{1s}(\mathbf{a}, \mathbf{b})$.

```

1 import math
2 import numpy as np
3 import scipy.integrate as sp
4 from scipy.special import spherical_in, spherical_kn, eval_legendre
5
6 def J1s(l, r, a): # Explicit J1s expression for hydrogen 1s orbital
7     (rL, rG) = (np.fmin(r, a), np.fmax(r, a))
8     A = rG * spherical_kn(max(l-1,0), rG) * spherical_in(l, rL)
9     B = rL * spherical_kn(l, rG) * spherical_in(l+1, rL)
10    J = 4/math.pi * (A - B)
11    return J
12
13 # Matrix element <1s(a)| 1/r |1s(b)>
14 def h1s(a, b, cosTheta):
15     # Logarithmic grid for the radial integrals
16     (rmin, rmax, dx) = (1e-6, 50, 0.005)
17     r = np.exp(np.arange(math.log(rmin), math.log(rmax), dx))
18
19     # Sum of the radial integrals <Ja|1/r|Jb> = int dr r Ja Jb = int dx r^2 Ja Jb
20     res = 0
21     for l in range(100):
22         Ja = J1s(l, r, a)
23         Jb = J1s(l, r, b)
24         I = (2*l+1) * sp.simps(r**2*Ja*Jb, dx=dx)
25         res += I * eval_legendre(l, cosTheta)
26         if abs(I)<1e-6:
27             break
28     return res

```

9.4. Long-range Coulomb matrix elements

In this section, we consider the Coulomb elements with off-centers:

$$(\alpha(\mathbf{a})\beta(\mathbf{b})|\frac{1}{r_{12}}|\gamma(\mathbf{c})\delta(\mathbf{d})) \quad \text{where} \quad \frac{1}{r_{12}} = \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \quad (9.40)$$

It must be pointed out that, it is already difficult enough to compute a general on-site Coulomb matrix element (see Eqn. (4.10)):

$$U_{\alpha\beta\gamma\delta}^{(0)} = (\alpha\beta|\frac{1}{r_{12}}|\gamma\delta)$$

To work with the additional off-centering, here we simplify our discussion on spherically symmetric orbitals. We consider the three types:

$$\begin{aligned} U_{s_1 s_2}^{(1)}(\mathbf{a}) &= (s_1 s_2(\mathbf{a})|\frac{1}{r_{12}}|s_1 s_1) \\ U_{s_1 s_2}^{(d)}(\mathbf{a}) &= (s_1 s_2(\mathbf{a})|\frac{1}{r_{12}}|s_2(\mathbf{a})s_1) \\ U_{s_1 s_2}^{(x)}(\mathbf{a}) &= (s_1 s_2(\mathbf{a})|\frac{1}{r_{12}}|s_1 s_2(\mathbf{a})) \end{aligned}$$

All the three involve two s -orbitals, s_1 centered at the origin and s_2 centered at $\mathbf{a}$. $U_{s_1 s_2}^{(1)}$ has only one off-center; $U_{s_1 s_2}^{(d)}$, where d stands for direct, is a typical direct Coulomb integral; $U_{s_1 s_2}^{(x)}$, where x stands for exchange, is a typical exchange Coulomb integral. Since the spin-integrals are trivial in the Coulomb elements (4.10), we discuss only the spatial-integrals in this section.

First, we perform the spherical wave expansions (8.25) and the re-centerings (9.5):

$$U_{s_1 s_2}^{(1)}(\mathbf{a}) = \int dk k \overline{I_{s_2}(k)} \sum_{lm} \overline{g_{lm,s}(k\mathbf{a})} (s_1, klm|\frac{1}{r_{12}}|s_1 s_1) \quad (9.41)$$

$$U_{s_1 s_2}^{(d)}(\mathbf{a}) = \int dk \int dk' k k' \overline{I_{s_2}(k)} I_{s_2}(k') \sum_{lm,l'm'} \overline{g_{lm,s}(k\mathbf{a})} g_{l'm',s}(k'\mathbf{a}) (s_1, klm|\frac{1}{r_{12}}|k'l'm', s_1) \quad (9.42)$$

$$U_{s_1 s_2}^{(x)}(\mathbf{a}) = \int dk \int dk' k k' \overline{I_{s_2}(k)} I_{s_2}(k') \sum_{lm,l'm'} \overline{g_{lm,s}(k\mathbf{a})} g_{l'm',s}(k'\mathbf{a}) (s_1, klm|\frac{1}{r_{12}}|s_1, k'l'm') \quad (9.43)$$

where (from (8.24) and (9.6)),

$$I_{s_2}(k) = \sqrt{\frac{2}{\pi}} \int_0^\infty dr r^2 j_0(kr) R_{s_2}(r) \quad (9.44)$$

$$g_{lm,s}(k\mathbf{a}) = 4\pi \langle lm|lm|s \rangle j_l(ka) \overline{Y_{lm}}(\hat{\mathbf{a}}) = \sqrt{4\pi} j_l(ka) \overline{Y_{lm}}(\hat{\mathbf{a}}) \quad (9.45)$$

9. Re-centering method and multi-center matrix elements

However, since we are working with spherically symmetric orbitals, the Coulomb integral does not depend on the direction vector $\hat{\mathbf{a}}$. We can orient the coordinates such that $\hat{\mathbf{a}}$ points to the z -direction, thus the re-centering coefficient simplifies as in (9.8),

$$g_{lm,s}(k\mathbf{a}) = \sqrt{2l+1} j_l(ka) \delta_{m0} \quad (9.46)$$

Next, we massage the centered Coulomb integrals using the multipole expansion (4.11):

$$\begin{aligned} (s_1, kl0 | \frac{1}{r_{12}} | s_1 s_1) &= \sum_{\lambda\mu} \frac{4\pi}{2\lambda+1} (R_{s_1} R_{kl} | \frac{r_{<}^\lambda}{r_{>^{\lambda+1}}} | R_{s_1} R_{s_1}) \langle Y_s | \overline{Y_{\lambda\mu}} | Y_s \rangle \langle Y_{l0} | Y_{\lambda\mu} | Y_s \rangle \\ &= \delta_{l0} (R_{s_1} R_{ks} | \frac{1}{r_{>}} | R_{s_1} R_{s_1}) \end{aligned} \quad (9.47)$$

$$\begin{aligned} (s_1, kl0 | \frac{1}{r_{12}} | k'l'0, s_1) &= \sum_{\lambda\mu} \frac{4\pi}{2\lambda+1} (R_{s_1} R_{kl} | \frac{r_{<}^\lambda}{r_{>^{\lambda+1}}} | R_{k'l'} R_{s_1}) \langle Y_s | \overline{Y_{\lambda\mu}} | Y_s \rangle \langle Y_{l0} | Y_{\lambda\mu} | Y_{l'0} \rangle \\ &= \delta_{ll'} (R_{s_1} R_{kl} | \frac{1}{r_{>}} | R_{k'l} R_{s_1}) \end{aligned} \quad (9.48)$$

$$\begin{aligned} (s_1, kl0 | \frac{1}{r_{12}} | s_1, k'l'0) &= \sum_{\lambda\mu} \frac{4\pi}{2\lambda+1} (R_{s_1} R_{kl} | \frac{r_{<}^\lambda}{r_{>^{\lambda+1}}} | R_{s_1} R_{k'l'}) \langle Y_s | \overline{Y_{\lambda\mu}} | Y_{l'0} \rangle \langle Y_{l0} | Y_{\lambda\mu} | Y_s \rangle \\ &= \frac{\delta_{ll'}}{2l+1} (R_{s_1} R_{kl} | \frac{r_{<}^l}{r_{>^{l+1}}} | R_{s_1} R_{k'l}) \end{aligned} \quad (9.49)$$

Collecting the results:

$$U_{s_1 s_2}^{(1)}(\mathbf{a}) = (R_{s_1} J_{s_2}^{(0)}(r_2, a) | \frac{1}{r_{>}} | R_{s_1} R_{s_1}) \quad (9.50)$$

$$U_{s_1 s_2}^{(d)}(\mathbf{a}) = \sum_{l=0}^{\infty} (2l+1) (R_{s_1} J_{s_2}^{(l)}(r_2, a) | \frac{1}{r_{>}} | J_{s_2}^{(l)}(r_2, a) R_{s_1}) \quad (9.51)$$

$$U_{s_1 s_2}^{(x)}(\mathbf{a}) = \sum_{l=0}^{\infty} (R_{s_1} J_{s_2}^{(l)}(r_2, a) | \frac{r_{<}^l}{r_{>^{l+1}}} | R_{s_1} J_{s_2}^{(l)}(r_1, a)) \quad (9.52)$$

where $J_{\alpha}^{(l)}$ is defined in (9.28). Thus, the 6-dimensional ($\mathbf{r}_1$ and $\mathbf{r}_2$) long-range Coulomb integrals reduce to sums of 2-dimensional (r_1 and r_2) radial integrals.

As an example, we consider the hydrogen $1s$ orbital ($s_1 = s_2 = H_{1s}$). We have worked out $J_{1s}^{(l)}$ in (9.39). Substituting $J_{1s}^{(l)}$ and R_{1s} back to (9.50), (9.51), and (9.52), we obtain

the long-range Coulomb integrals explicitly:

$$\begin{aligned}
U_{1s}^{(1)}(\mathbf{a}) &= \int_0^\infty dr_2 \, r_2 J_{1s}^{(0)}(r_2, a) u_{1s}(r_2) \int_0^\infty dr_1 \, u_{1s}^2(r_1) \frac{1}{r_{>}} \\
&= \int_0^\infty dr_2 \, r_2 J_{1s}^{(0)}(r_2, a) u_{1s}(r_2) \left(\int_0^{r_2} dr_1 \, \frac{u_{1s}^2(r_1)}{r_2} + \int_{r_2}^\infty dr_1 \, \frac{u_{1s}^2(r_1)}{r_1} \right) \\
&= \int_0^\infty dr_2 \, r_2 J_{1s}^{(0)}(r_2, a) u_{1s}(r_2) \left(\frac{1 - (r_2 + 1)e^{-2r_2}}{r_2} \right) \\
&= \int_0^\infty dr \, J_{1s}^{(0)}(r, a) [2re^{-r} - 2(r^2 + r)e^{-3r}] \\
&= \frac{4}{\pi} \int_0^a dr \, [ak_{-1}(a)i_0^{(1)}(r) - rk_0(a)i_1^{(1)}(r)] [2re^{-r} - 2(r^2 + r)e^{-3r}] \\
&\quad + \frac{4}{\pi} \int_a^\infty dr \, [rk_{-1}(r)i_0^{(1)}(a) - ak_0(r)i_1^{(1)}(a)] [2re^{-r} - 2(r^2 + r)e^{-3r}] \quad (9.53)
\end{aligned}$$

From (DLMF 10.49.9), (DLMF 10.49.13), and (DLMF 10.47.9) [54],

$$i_0^{(1)}(z) = \frac{\sinh z}{z}, \quad i_1^{(1)}(z) = -\frac{\sinh z}{z^2} + \frac{\cosh z}{z}, \quad k_{-1}(z) = k_0(z) = \frac{\pi}{2} \frac{e^{-z}}{z} \quad (9.54)$$

We obtain the explicit result,

$$U_{1s}^{(1)}(\mathbf{a}) = ae^{-a} + \left(\frac{1}{8} + \frac{5}{16a} \right) (e^{-a} - e^{-3a}) \quad (9.55)$$

$U_{1s}^{(d)}$ reduces to a sum of 1-dimensional radial integrals,

$$\begin{aligned}
U_{1s}^{(d)}(\mathbf{a}) &= \sum_{l=0}^{\infty} (2l+1) \int_0^\infty dr_2 \, r_2^2 |J_{1s}^{(l)}(r_2, a)|^2 \int_0^\infty dr_1 \, u_{1s}^2(r_1) \frac{1}{r_{>}} \\
&= \sum_{l=0}^{\infty} (2l+1) \int_0^\infty dr_2 \, r_2^2 |J_{1s}^{(l)}(r_2, a)|^2 \left(\frac{1 - (r_2 + 1)e^{-2r_2}}{r_2} \right) \\
&= \sum_{l=0}^{\infty} (2l+1) \left\langle J_{1s}^{(l)}(r, a) \left| \frac{1 - (r+1)e^{-2r}}{r} \right| J_{1s}^{(l)}(r, a) \right\rangle \quad (9.56)
\end{aligned}$$

Algorithm 9.2 provides example routines for computing $U_{1s}^{(d)}(\mathbf{a})$ and $U_{1s}^{(x)}(\mathbf{a})$. To be consistent with our earlier discussions, we compute the integrals numerically on the logarithmic grid (3.12). We re-use the `cumsimps` function defined in Algorithm 3.6.

Algorithm 9.2: Routines for computing $U_{1s}^{(d)}(\mathbf{a})$ and $U_{1s}^{(x)}(\mathbf{a})$. The `cumsimps` function is defined in Algorithm 3.6 and the `J1s` function is given in Algorithm 9.1.

```

1 import math
2 import numpy as np
3 import scipy.integrate as sp
4

```

9. Re-centering method and multi-center matrix elements

```

5 def U1s_d(a):
6     # Logarithmic grid for the radial integrals (dr = r*dx)
7     (rmin, rmax, dx) = (1e-6, 50, 0.005)
8     r = np.exp(np.arange(math.log(rmin), math.log(rmax), dx))
9     F = r**2 * (1-(r+1)*np.exp(-2*r)) # F = r**3 * f
10
11     # Sum of radial integrals <J|f|J> = int dr r**2 f J^2 = int dx r**3 f J^2
12     res = 0
13     for l in range(100):
14         J = J1s(l, r, a)
15         I = (2*l+1) * sp.simps(J**2*F, dx=dx)
16         res += I
17         if abs(I)<1e-6:
18             break
19     return res
20
21 def U1s_x(a):
22     # Logarithmic grid for the radial integrals (dr = r*dx)
23     (rmin, rmax, dx) = (1e-6, 50, 0.005)
24     r = np.exp(np.arange(math.log(rmin), math.log(rmax), dx))
25     F = r**3 * 2*np.exp(-r) # F = r**3 * R
26
27     # Sum of radial integrals (R,J|r<^l/r>^{l+1}|R,J)
28     res = 0
29     for l in range(100):
30         J = J1s(l, r, a)
31         A = cumsimps( J*F*r**l, dx)
32         B = cumsimps((J*F/r**(l+1))[:-1], dx)[:-1]
33         I = sp.simps( J*F*(A/r**(l+1)+B*r**l), dx=dx)
34         res += I
35         if abs(I)<1e-6:
36             break
37     return res

```

At this stage, we summarize the multi-center integrals including, overlap, hopping, and Coulomb elements, in Table 9.1. In particular, the multi-center elements for hydrogen 1s-orbitals are summarized in Table 9.2. The corresponding plots are shown in Fig. 9.5.

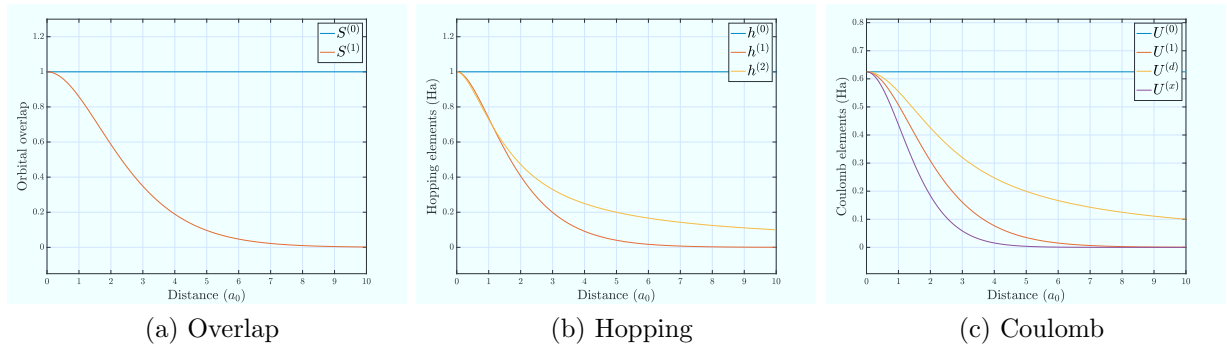


Figure 9.5.: Multi-center (overlap, hopping, Coulomb) integrals for hydrogen 1s-orbitals.

Table 9.1.: Multi-center (overlap, hopping, Coulomb) integrals.

$$\begin{aligned}
 I_\alpha(k) &\equiv \sqrt{\frac{2}{\pi}} \int_0^\infty dr \, r^2 j_{l_\alpha}(kr) R_\alpha(r) \\
 I_{\tilde{\alpha}}(k) &= \sqrt{\frac{2}{\pi}} \int_0^\infty dr \, r j_{l_\alpha}(kr) R_\alpha(r) \\
 g_{\alpha\beta}(k\mathbf{a}) &\equiv 4\pi \sum_{lm} i^{l_\alpha - l_\beta - l} \langle Y_\alpha | Y_{lm} | Y_\beta \rangle j_l(ka) \overline{Y_{lm}}(\hat{\mathbf{a}}) \\
 J_\alpha^{(l)}(r, a) &\equiv \sqrt{\frac{2}{\pi}} \int_0^\infty dk \, k^2 I_\alpha(k) j_l(kr) j_l(ka)
 \end{aligned}$$

$S_{\alpha\beta}^{(0)}$	$=$	$\langle \alpha \beta \rangle$	
$S_{\alpha\beta}^{(1)}(\mathbf{a})$	$=$	$\langle \alpha \beta(\mathbf{a}) \rangle$	$= \int_0^\infty dk \, k^2 \overline{I_\alpha}(k) I_\beta(k) g_{\alpha\beta}(k\mathbf{a})$
$h_{\alpha\beta}^{(0)}$	$=$	$\langle \alpha \frac{1}{r} \beta \rangle$	
$h_{\alpha\beta}^{(1)}(\mathbf{a})$	$=$	$\langle \alpha \frac{1}{r} \beta(\mathbf{a}) \rangle$	$= \int_0^\infty dk \, k^2 \overline{I_\alpha}(k) I_\beta(k) g_{\alpha\beta}(k\mathbf{a})$
$h_{\alpha\beta}^{(2)}(\mathbf{a})$	$=$	$\langle \alpha(\mathbf{a}) \frac{1}{r} \beta(\mathbf{a}) \rangle$	$= \sum_{l=0}^\infty \frac{4\pi}{2l+1} \langle R_\alpha \frac{r_{\leq}^l}{r_{>}^{l+1}} R_\beta \rangle \sum_{m=-l}^l \langle Y_\alpha Y_{lm} Y_\beta \rangle \overline{Y_{lm}}(-\hat{\mathbf{a}})$
$h_{s_1 s_2}(\mathbf{a}, \mathbf{b})$	$=$	$\langle s_1(\mathbf{a}) \frac{1}{r} s_2(\mathbf{b}) \rangle$	$= \sum_{l=0}^\infty (2l+1) \left\langle J_{s_1}^{(l)}(r, a) \left \frac{1}{r} \right J_{s_2}^{(l)}(r, b) \right\rangle P_l(\hat{\mathbf{a}} \cdot \hat{\mathbf{b}})$
$U_{\alpha\beta\gamma\delta}^{(0)}$	$=$	$(\alpha\beta \frac{1}{r_{12}} \gamma\delta)$	
$U_{s_1 s_2}^{(1)}(\mathbf{a})$	$=$	$(s_1 s_2(\mathbf{a}) \frac{1}{r_{12}} s_1 s_1)$	$= \left(R_{s_1} J_{s_2}^{(0)}(r_2, a) \left \frac{1}{r_{>}} \right R_{s_1} R_{s_1} \right)$
$U_{s_1 s_2}^{(d)}(\mathbf{a})$	$=$	$(s_1 s_2(\mathbf{a}) \frac{1}{r_{12}} s_2(\mathbf{a}) s_1)$	$= \sum_{l=0}^\infty (2l+1) \left(R_{s_1} J_{s_2}^{(l)}(r_2, a) \left \frac{1}{r_{>}} \right J_{s_2}^{(l)}(r_2, a) R_{s_1} \right)$
$U_{s_1 s_2}^{(x)}(\mathbf{a})$	$=$	$(s_1 s_2(\mathbf{a}) \frac{1}{r_{12}} s_1 s_2(\mathbf{a}))$	$= \sum_{l=0}^\infty \left(R_{s_1} J_{s_2}^{(l)}(r_2, a) \left \frac{r_{\leq}^l}{r_{>}^{l+1}} \right R_{s_1} J_{s_2}^{(l)}(r_1, a) \right)$

9. Re-centering method and multi-center matrix elements

Table 9.2.: Multi-center integrals in Table 9.1 applied on hydrogen 1s-orbitals.

$$J_{1s}^{(l)}(r, a) = \frac{4}{\pi} \left[r_{>} k_{l-1}(r_{>}) i_l^{(1)}(r_{<}) - r_{<} k_l(r_{>}) i_{l+1}^{(1)}(r_{<}) \right]$$

$S_{1s}^{(0)} = 1$ $S_{1s}^{(1)}(\mathbf{a}) = \frac{1}{3}(a^2 + 3a + 3)e^{-a}$
$h_{1s}^{(0)} = 1$ $h_{1s}^{(1)}(\mathbf{a}) = (a + 1)e^{-a}$ $h_{1s}^{(2)}(\mathbf{a}) = \frac{1}{a} - \left(1 + \frac{1}{a}\right)e^{-2a}$ $h_{1s}(\mathbf{a}, \mathbf{b}) = \sum_{l=0}^{\infty} (2l + 1) \left\langle J_{1s}^{(l)}(r, a) \left \frac{1}{r} \right J_{1s}^{(l)}(r, b) \right\rangle P_l(\hat{\mathbf{a}} \cdot \hat{\mathbf{b}})$
$U_{1s}^{(0)} = \frac{5}{8}$ $U_{1s}^{(1)}(\mathbf{a}) = ae^{-a} + \left(\frac{1}{8} + \frac{5}{16a}\right)(e^{-a} - e^{-3a})$ $U_{1s}^{(d)}(\mathbf{a}) = \sum_{l=0}^{\infty} (2l + 1) \left\langle J_{1s}^{(l)}(r, a) \left \frac{1 - (r + 1)e^{-2r}}{r} \right J_{1s}^{(l)}(r, a) \right\rangle$ $U_{1s}^{(x)}(\mathbf{a}) = \sum_{l=0}^{\infty} \left(R_{1s} J_{1s}^{(l)}(r_2, a) \left \frac{r_{<}^l}{r_{>}^{l+1}} \right R_{1s} J_{1s}^{(l)}(r_1, a) \right)$

9.5. Basis orthonormalization effects on the matrix elements

The general one- and two- body Hamiltonians (1.15) in their second quantization forms are most convenient to work with if the chosen basis is orthonormal. In this section, we assume that the given basis is atomic (non-orthogonal), and we are interested in understanding the modifications on the matrix elements after the basis orthonormalization is performed.

Suppose we are given the one- and two- body matrix elements in an atomic basis:

$$h_{ij}^{\varphi} \quad \text{and} \quad v_{ijkl}^{\varphi}$$

Meanwhile, the basis orthonormalization is given by the linear transformation:

$$|\perp_{\alpha}\rangle = \sum_i |\varphi_i\rangle T_{i\alpha}$$

Under the basis transformation, the matrix elements are transformed accordingly. For instance, consider the two transformed orbitals (8.58) and (8.59). We have,

$$\begin{aligned} \langle \perp_1 | \frac{1}{r} | \perp_2 \rangle &= \left(c_1 \langle \varphi_1 | + c_2 \langle \varphi_2 | \right) \frac{1}{r} \left(c_2 | \varphi_1 \rangle + c_1 | \varphi_2 \rangle \right) \\ &= c_1 c_2 \langle \varphi_1 | \frac{1}{r} | \varphi_1 \rangle + c_1^2 \langle \varphi_1 | \frac{1}{r} | \varphi_2 \rangle + c_2^2 \langle \varphi_2 | \frac{1}{r} | \varphi_1 \rangle + c_2 c_1 \langle \varphi_2 | \frac{1}{r} | \varphi_2 \rangle \end{aligned}$$

In general, the transformations of the matrix elements under the two basis can be obtained by [6],

$$h_{\alpha\beta}^{\perp} = \sum_{ij} T_{\alpha i}^{\dagger} h_{ij}^{\varphi} T_{j\beta} \quad (9.57)$$

$$v_{\alpha\beta\gamma\delta}^{\perp} = \sum_{ijkl} T_{\alpha i}^{\dagger} T_{\beta j}^{\dagger} v_{ijkl}^{\varphi} T_{k\gamma} T_{l\delta} \quad (9.58)$$

An example code for transforming one-body and two-body elements is given in Algorithm 9.3.

Algorithm 9.3: Basis transformations of matrix elements.

```

1 import numpy as np
2
3 # Basis transformation of a one-body matrix
4 def ThT(h, T):
5     return T.conj().transpose() @ h @ T
6
7 # Basis transformation of a two-body tensor
8 def TTvTT(v, T):
9     Th = T.conj().transpose()
10    vt = np.zeros_like(v)
11
12    orbs = range(len(T))
13    idx = [(a,b,c,d) for a in orbs for b in orbs for c in orbs for d in orbs]
14    for (a,b,c,d) in idx:
15        for (i,j,k,l) in idx:
16            vt[a,b,c,d] += Th[a,i]*Th[b,j]*v[i,j,k,l]*T[k,c]*T[l,d]
17    return vt

```

9. Re-centering method and multi-center matrix elements

The main interest in the section is to understand the modification on the matrix elements due to the deformation of the orbitals. We first study the multi-center matrix elements of the two-site model consisting of two hydrogen 1s-orbitals under the original atomic basis and the Löwdin orthogonalized basis. The resulting matrix elements are plotted in Fig. 9.6. The dashed lines (copied from Fig. 9.5) are the matrix elements under the original atomic basis; The solid lines are the matrix elements under the Löwdin orthogonalized basis.

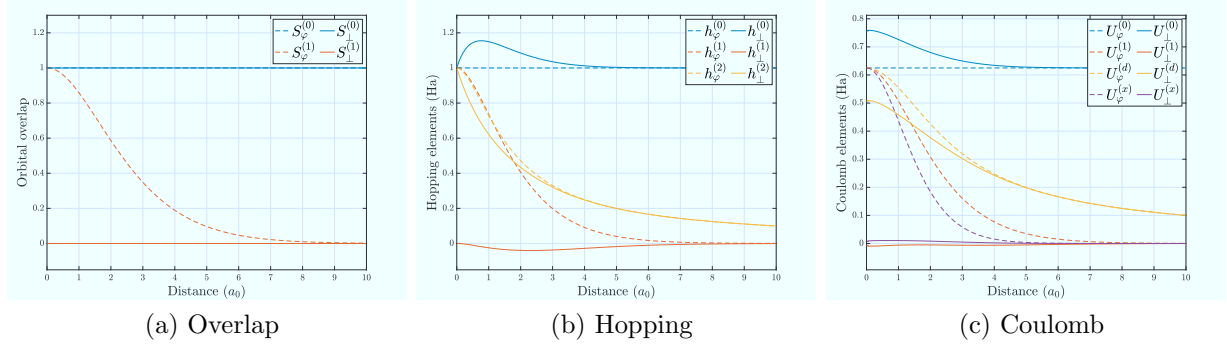


Figure 9.6.: Transformed multi-center (overlap, hopping, Coulomb) integrals of a two-site system consisting of two hydrogen 1s-orbitals. The dashed lines (copied from Fig. 9.5) are the matrix elements under the original atomic basis; The solid lines are the matrix elements under the Löwdin orthogonalized basis.

Fig. 9.6a shows the orbital overlaps. After the orthonormalization, the on-site overlap stays at 1 (normalized) and the off-site overlap becomes 0 (orthogonalized).

Fig. 9.6b plots the hopping elements. After the orthonormalization, the biggest change is on the element $h^{(1)}$, which almost vanishes. The element $h^{(1)} = \langle \alpha | \frac{1}{r} | \beta(\mathbf{a}) \rangle$ is closely related to the overlap $S^{(1)} = \langle \alpha | \beta(\mathbf{a}) \rangle$. They both decay exponentially as a function of the separation due to the decaying wave functions. On the other hand, the element $h^{(2)}$ is not significantly modified. Notice that the element $h^{(2)} = \langle \alpha(\mathbf{a}) | \frac{1}{r} | \beta(\mathbf{a}) \rangle$ is basically an on-site element in the environment of an off-site potential. The decaying behavior of $h^{(2)}$ is simply $1/r$ as the potential goes. Under the Löwdin orthonormalization, the original orbitals are minimally deformed. As a result, $h^{(2)}$ are only slightly modified.

Fig. 9.6c plots the Coulomb elements. The behaviors of the matrix elements are similar to the hopping elements. As an effect of the orthogonalization, the $U^{(1)}$ and $U^{(x)}$ are reduced to near zero values. Only the on-site $U^{(0)}$ and the direct Coulomb element $U^{(d)}$ stay minimally modified. The on-site $U^{(0)}$ increases when the two sites get closer. The value grows up because the orthogonalized orbitals are more compressed and localized (see Fig. 8.7). A more compressed orbital means a higher on-site Coulomb energy. On the other hand, the direct Coulomb element $U^{(d)}$ is slightly reduced when the two sites get closer. The value reduces because the orbitals after orthogonalization tend to avoid each other, and the centers of the gravity are relatively pushed away, which results in a lower value of the direct Coulomb element.

9.5. Basis orthonormalization effects on the matrix elements

The three plots in Fig. 9.6 presents the effects of the orthonormalization on the matrix elements of a simple two-site model. Certainly, for systems with different configurations, the orthonormalization scheme produces different orthonormal orbitals (e.g. Fig. 8.8) and the corresponding matrix elements would be different. To understand the system dependence, we study the hopping matrix elements of different ring systems under the orthogonalization scheme. Table 9.3 gives a list of ring systems consisting of hydrogen $1s$ -orbitals with the nearest-neighbor distance $a = 2 a_0$ ranging from two sites to six sites. Essentially, we see that modification on the matrix elements does not have a strong dependence on the number of sites. The general behaviors of the multi-center elements and the effects of the orthonormalization are essentially captured in the two-site model.

Table 9.3.: Transformed matrix elements under the effect of the Löwdin orthonormalization with different system configurations. All the ring systems consist of hydrogen $1s$ -orbitals with the nearest-neighbor distance $a = 2 a_0$. The matrix elements are given in units of Hartree.

	Original					
$h_{\alpha\beta}^{(0)}$	1.000	1.085	1.126	1.166	1.184	1.198
$h_{\alpha\beta}^{(1)}$	0.406	-0.039	-0.020	-0.040	-0.044	-0.053
$h_{\alpha\beta}^{(2)}$	0.473	0.434	0.432	0.443	0.452	0.458

10. Summary

If you run a simple line of code,

```
print([format(i,'06b') for i in range(1<<6) if bin(i).count('1')==2])
```

you have basically created the occupation representation of a many-electron basis.

The line above generates 15 binary strings, which represent the 15 many-electron basis configurations of a system with 6 basis-orbitals and 2 electrons. However, you soon realize that, if you increase the system size, the basis dimension will explode exponentially and eventually kill your laptop (try 30 orbitals and 15 electrons).

Simply enumerating the basis configuration is already hard. Solving the many-electron Hamiltonians brute-force is practically impossible. Nevertheless, Chapter 1 provides explicit algorithms for setting up many-electron bases and matrix representations of Hamiltonians. It sets up a stage where we can use the *exact diagonalization* method to solve relatively simple systems and compare results with other methods.

We first discussed in Chapter 2 the tight-binding method to solve and understand realistic materials. Here we discussed the basic concepts of the Bloch waves and energy bands. However, building a realistic model requires having the realistic basis orbitals. In Chapter 3, we developed an SCF solver to compute many-electron atomic systems *ab-initio*. The resulting self-consistent atomic orbitals and potentials are the building blocks for modeling any material. Normally, when people talk about DFT, they think about heavy calculations with some million-line-code software. Here, within three hundred lines, we developed a nice and compact DFT code. In fact, we have taught the code in one of our master's courses, and received very positive feedbacks from the students that they understood the problems into detail and comprehended the DFT calculations more intuitively.

In Chapter 4, we discussed solving many-electron atomic open-shell systems using the *LS*- and *jj*-coupling schemes. Atomic open-shells are precious many-electron systems that can be solved *analytically*. The resulting many-electron eigen-states and eigen-energies are known as the multiplets. Multiplets present many-electron properties of materials in the atomic limit. In Chapter 5, we asked a question if there is a pattern among the matrix elements across the periodic table. We studied systematically for atomic systems the trends of the Coulomb and spin-orbit interactions and the corresponding matrix elements. We provided the trends of values as fit functions for practical calculations.

The comparison of the interaction strengths from Chapter 5 gave rise to the study of the moment formulas in Chapter 6. Here we tried – in some sense – to solve the general

10. Summary

many-body problem. We solved a general one-body Hamiltonian up to an arbitrary moment; we also solved a general two-body Hamiltonian up to the 2nd moment. The moment formulas give us exact solutions of many-electron systems without the need of working with a many-electron basis. Certain physical information of the many-electron system (e.g. the spectral variance), which could cost terabytes of computer memories and hundreds of computational hours from a standard many-body approach, can be now computed *analytically*. To derive the moment formulas, a lot of techniques and tricks such as the density matrix expansion and the conditional combinatorics are used. However, a good physicist should have already smelled that there is something deep here. As we saw that the difficulty of the problems explodes quickly during the derivations, but collapses suddenly to some compact forms in the end. From lower to higher moments, the solutions exhibit *beautiful patterns*. We believe that it is extremely valuable to dig deeper into this topic and eventually discover more symmetries and general solutions for the many-body problems.

Chapter 7 is needed for discussing the multi-center integrals in the next two chapters. Here we discussed the integral of three spherical harmonics – the Gaunt coefficient. Such integrals are ubiquitous for problems that involve spherical harmonic expansions. For expansions that involve large angular momenta, computing the corresponding Gaunt numbers accurately can be challenging. We developed a recursive algorithm based on finite-precision arithmetic to compute the Gaunt coefficients efficiently and accurately.

In Chapter 8, we discussed a systematic way to compute orbital overlaps and introduced the overlap formula. The overlap formula is general for evaluating overlaps involving any atomic orbitals with arbitrary relative displacements. Given a non-orthonormal basis set, we can compute the overlap matrix and produce a new set of basis orbitals which are mutually orthonormal, yet preserving their atomic features as much as possible using the Löwdin symmetric orthogonalization scheme. In Chapter 9, we generalized the multi-center problems using the re-centering method. We studied the evaluation of general multi-center integrals including hopping matrix elements and long-range Coulomb matrix elements, and the effects of orbital orthogonalization on the resulting matrix elements.

From density functional theory to many-body methods, from intensive numerical calculations to heavy analytical derivations, our results provide a solid basis for simulating realistic many-electron systems. The developed algorithms and methods – in particular the moment approach – opens possibility and brings new insights for understanding the very challenging quantum many-body problems.

A. Hydrogen reference tables

A.1. Hydrogen-like radial wave functions

Table A.1.: Radial wave functions $u_{nl}(r)$ ($n = 1, 2, 3, 4$) for hydrogen-like atoms with atomic number Z . A rescaled coordinate $\rho = Zr$ (r in units of Bohr radius) is used to simplify the expressions. Eigen-energies: $E_n = -\frac{Z^2}{2n^2}$ (Hartree).

$u_{10} =$	$2 \sqrt{Z} \rho$	$\exp(-\rho)$
$u_{20} =$	$\frac{1}{\sqrt{2}} \sqrt{Z} \rho \left(1 - \frac{1}{2}\rho\right)$	$\exp(-\rho/2)$
$u_{21} =$	$\frac{1}{\sqrt{24}} \sqrt{Z} \rho^2$	$\exp(-\rho/2)$
$u_{30} =$	$\frac{2}{\sqrt{27}} \sqrt{Z} \rho \left(1 - \frac{2}{3}\rho + \frac{2}{27}\rho^2\right)$	$\exp(-\rho/3)$
$u_{31} =$	$\frac{8}{27\sqrt{6}} \sqrt{Z} \rho^2 \left(1 - \frac{1}{6}\rho\right)$	$\exp(-\rho/3)$
$u_{32} =$	$\frac{4}{81\sqrt{30}} \sqrt{Z} \rho^3$	$\exp(-\rho/3)$
$u_{40} =$	$\frac{1}{4} \sqrt{Z} \rho \left(1 - \frac{3}{4}\rho + \frac{1}{8}\rho^2 - \frac{1}{192}\rho^3\right)$	$\exp(-\rho/4)$
$u_{41} =$	$\frac{5}{16\sqrt{15}} \sqrt{Z} \rho^2 \left(1 - \frac{1}{4}\rho + \frac{1}{80}\rho^2\right)$	$\exp(-\rho/4)$
$u_{42} =$	$\frac{1}{64\sqrt{5}} \sqrt{Z} \rho^3 \left(1 - \frac{1}{12}\rho\right)$	$\exp(-\rho/4)$
$u_{43} =$	$\frac{1}{768\sqrt{35}} \sqrt{Z} \rho^4$	$\exp(-\rho/4)$

A. Hydrogen reference tables

Table A.1 extended for $n = 5, 6, 7$.

$u_{50} =$	$\frac{2}{5\sqrt{5}}$	$\sqrt{Z}\rho$	$\left(1 - \frac{4}{5}\rho + \frac{4}{25}\rho^2 - \frac{4}{375}\rho^3 + \frac{2}{9375}\rho^4\right)$	$\exp(-\rho/5)$
$u_{51} =$	$\frac{8}{25\sqrt{30}}$	$\sqrt{Z}\rho^2$	$\left(1 - \frac{3}{10}\rho + \frac{3}{125}\rho^2 - \frac{1}{1875}\rho^3\right)$	$\exp(-\rho/5)$
$u_{52} =$	$\frac{28}{625\sqrt{70}}$	$\sqrt{Z}\rho^3$	$\left(1 - \frac{2}{15}\rho + \frac{2}{525}\rho^2\right)$	$\exp(-\rho/5)$
$u_{53} =$	$\frac{16}{9375\sqrt{70}}$	$\sqrt{Z}\rho^4$	$\left(1 - \frac{1}{20}\rho\right)$	$\exp(-\rho/5)$
$u_{54} =$	$\frac{4}{140625\sqrt{70}}$	$\sqrt{Z}\rho^5$		$\exp(-\rho/5)$
$u_{60} =$	$\frac{1}{3\sqrt{6}}$	$\sqrt{Z}\rho$	$\left(1 - \frac{5}{6}\rho + \frac{5}{27}\rho^2 - \frac{5}{324}\rho^3 + \frac{1}{1944}\rho^4 - \frac{1}{174960}\rho^5\right)$	$\exp(-\rho/6)$
$u_{61} =$	$\frac{35}{54\sqrt{210}}$	$\sqrt{Z}\rho^2$	$\left(1 - \frac{1}{3}\rho + \frac{1}{30}\rho^2 - \frac{1}{810}\rho^3 + \frac{1}{68040}\rho^4\right)$	$\exp(-\rho/6)$
$u_{62} =$	$\frac{7}{162\sqrt{105}}$	$\sqrt{Z}\rho^3$	$\left(1 - \frac{1}{6}\rho + \frac{1}{126}\rho^2 - \frac{1}{9072}\rho^3\right)$	$\exp(-\rho/6)$
$u_{63} =$	$\frac{1}{972\sqrt{35}}$	$\sqrt{Z}\rho^4$	$\left(1 - \frac{1}{12}\rho + \frac{1}{648}\rho^2\right)$	$\exp(-\rho/6)$
$u_{64} =$	$\frac{1}{104976\sqrt{7}}$	$\sqrt{Z}\rho^5$	$\left(1 - \frac{1}{30}\rho\right)$	$\exp(-\rho/6)$
$u_{65} =$	$\frac{1}{3149280\sqrt{77}}$	$\sqrt{Z}\rho^6$		$\exp(-\rho/6)$
$u_{70} =$	$\frac{2}{7\sqrt{7}}$	$\sqrt{Z}\rho$	$\left(1 - \frac{6}{7}\rho + \frac{10}{49}\rho^2 - \frac{20}{1029}\rho^3 + \frac{2}{2401}\rho^4 - \frac{4}{252105}\rho^5 + \frac{4}{37059435}\rho^6\right)$	$\exp(-\rho/7)$
$u_{71} =$	$\frac{8}{49\sqrt{21}}$	$\sqrt{Z}\rho^2$	$\left(1 - \frac{5}{14}\rho + \frac{2}{49}\rho^2 - \frac{2}{1029}\rho^3 + \frac{2}{50421}\rho^4 - \frac{1}{3529470}\rho^5\right)$	$\exp(-\rho/7)$
$u_{72} =$	$\frac{12}{343\sqrt{105}}$	$\sqrt{Z}\rho^3$	$\left(1 - \frac{4}{21}\rho + \frac{4}{343}\rho^2 - \frac{2}{7203}\rho^3 + \frac{1}{453789}\rho^4\right)$	$\exp(-\rho/7)$
$u_{73} =$	$\frac{16}{16807\sqrt{42}}$	$\sqrt{Z}\rho^4$	$\left(1 - \frac{3}{28}\rho + \frac{1}{294}\rho^2 - \frac{1}{30870}\rho^3\right)$	$\exp(-\rho/7)$
$u_{74} =$	$\frac{44}{1058841\sqrt{154}}$	$\sqrt{Z}\rho^5$	$\left(1 - \frac{2}{35}\rho + \frac{2}{2695}\rho^2\right)$	$\exp(-\rho/7)$
$u_{75} =$	$\frac{8}{12353145\sqrt{231}}$	$\sqrt{Z}\rho^6$	$\left(1 - \frac{1}{42}\rho\right)$	$\exp(-\rho/7)$
$u_{76} =$	$\frac{4}{259416045\sqrt{3003}}$	$\sqrt{Z}\rho^7$		$\exp(-\rho/7)$

A.2. Slater-Condon parameters

Table A.2.: Slater-Condon parameters for the exact hydrogen wave functions with $n = 1, 2, 3, 4, 5, 6, 7$. Energies are given in units of Hartree (a.u.). For hydrogen-like wave functions, $F_Z^{(k)} = ZF_H^{(k)}$.

Shell	k	$F_H^{(k)}$
$1s$	0	5/8
$2s$	0	77/512
$2p$	0	93/512
	2	45/512
$3s$	0	17/256
$3p$	0	1987/27648
	2	995/27648
$3d$	0	793/9216
	2	2093/46080
	4	91/3072
$4s$	0	19541/524288
$4p$	0	20413/524288
	2	10445/524288
$4d$	0	22373/524288
	2	56553/2621440
	4	7749/524288
$4f$	0	26333/524288
	2	103275/3670016
	4	69003/3670016
	6	7293/524288

Shell	k	$F_H^{(k)}$
$6s$	0	624353/37748736
$6p$	0	1909283/113246208
	2	995155/113246208
$6d$	0	2646769/150994944
	2	6790501/754974720
	4	317819/50331648
$6f$	0	937403/50331648
	2	10013975/1056964608
	4	20939237/3170893824
	6	773201/150994944
$6g$	0	9211115/452984832
	2	534611/50331648
	4	29887715/4076863488
	6	2554123/452984832
	8	2073643/452984832
$6h$	0	1172755/50331648
	2	2607787/184549376
	4	48790105/4982833152
	6	4085185/553648128
	8	3254225/553648128
	10	734825/150994944

Shell	k	$F_H^{(k)}$
$5s$	0	39043/1638400
$5p$	0	2007/81920
	2	1039/81920
$5d$	0	53111/2048000
	2	5403/409600
	4	18819/2048000
$5f$	0	186211/6553600
	2	133821/9175040
	4	460449/45875200
	6	251823/32768000
$5g$	0	43191/1310720
	2	75889/3932160
	4	31031/2359296
	6	193193/19660800
	8	51051/6553600

Shell	k	$F_H^{(k)}$
$7s$	0	1248305/102760448
$7p$	0	2893591/234881024
	2	10602465/1644167168
$7d$	0	145867055/11509170176
	2	376839867/57545850880
	4	53141031/11509170176
$7f$	0	38091791/2877292544
	2	19446615/2877292544
	4	13646039/2877292544
	6	10648495/2877292544
$7g$	0	40483999/2877292544
	2	62054677/8631877632
	4	1426795669/284851961856
	6	369613439/94950653952
	8	9185423/2877292544
$7h$	0	25152233/1644167168
	2	147116723/18085838848
	4	101801947/18085838848
	6	78546715/18085838848
	8	447928325/126600871936
	10	4900233/1644167168
$7i$	0	28539857/1644167168
	2	231477565/21374173184
	4	1789828119/235115905024
	6	1359198535/235115905024
	8	98868685/21374173184
	10	82083021/21374173184
	12	5386025/1644167168

A.3. Spin-orbit parameters

Table A.3.: Spin-orbit parameters for the exact hydrogen wave functions with $n = 2, 3, 4, 5, 6, 7$. Energies are given in units of Hartree (a.u.). In atomic units, $c \approx 137.036 \text{ a}_0/t_0$. For s -shells, there is no spin-orbit interaction. For hydrogen-like wave functions, $\Xi_Z = Z^4 \Xi_H$.

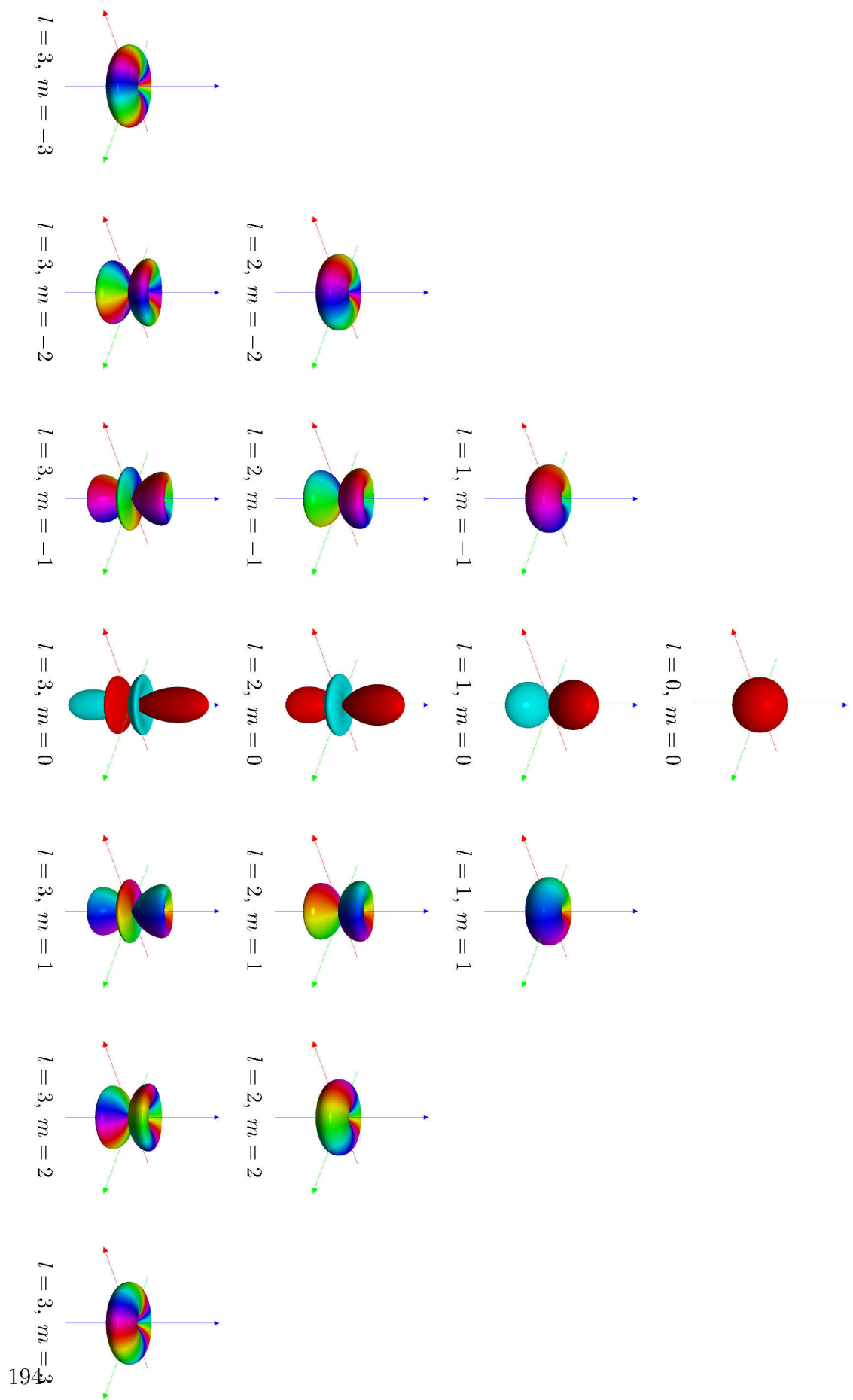
Shell	Ξ_H
$2p$	$1/(48c^2)$
$3p$	$1/(162c^2)$
$3d$	$1/(810c^2)$
$4p$	$1/(384c^2)$
$4d$	$1/(1920c^2)$
$4f$	$1/(5376c^2)$
$5p$	$1/(750c^2)$
$5d$	$1/(3750c^2)$
$5f$	$1/(10500c^2)$
$5g$	$1/(22500c^2)$
$6p$	$1/(1296c^2)$
$6d$	$1/(6480c^2)$
$6f$	$1/(18144c^2)$
$6g$	$1/(38880c^2)$
$6h$	$1/(71280c^2)$
$7p$	$1/(2058c^2)$
$7d$	$1/(10290c^2)$
$7f$	$1/(28812c^2)$
$7g$	$1/(61740c^2)$
$7h$	$1/(113190c^2)$
$7i$	$1/(187278c^2)$

A.4. The first few spherical harmonics

Table A.4.: The first few spherical harmonics.

$Y_{0,0} =$	$\sqrt{\frac{1}{4\pi}}$	
$Y_{1,0} =$	$\sqrt{\frac{3}{4\pi}} \cos \theta$	
$Y_{1,\pm 1} =$	$\mp \sqrt{\frac{3}{8\pi}} \sin \theta$	$e^{\pm i\phi}$
$Y_{2,0} =$	$\sqrt{\frac{5}{16\pi}} (3 \cos^2 \theta - 1)$	
$Y_{2,\pm 1} =$	$\mp \sqrt{\frac{15}{8\pi}} \sin \theta \cos \theta$	$e^{\pm i\phi}$
$Y_{2,\pm 2} =$	$\sqrt{\frac{15}{32\pi}} \sin^2 \theta$	$e^{\pm 2i\phi}$
$Y_{3,0} =$	$\sqrt{\frac{7}{16\pi}} (5 \cos^3 \theta - 3 \cos \theta)$	
$Y_{3,\pm 1} =$	$\mp \sqrt{\frac{21}{64\pi}} \sin \theta (5 \cos^2 \theta - 1)$	$e^{\pm i\phi}$
$Y_{3,\pm 2} =$	$\sqrt{\frac{105}{32\pi}} \sin^2 \theta \cos \theta$	$e^{\pm 2i\phi}$
$Y_{3,\pm 3} =$	$\mp \sqrt{\frac{35}{64\pi}} \sin^3 \theta$	$e^{\pm 3i\phi}$

Figure A.1.: The first few spherical harmonics.

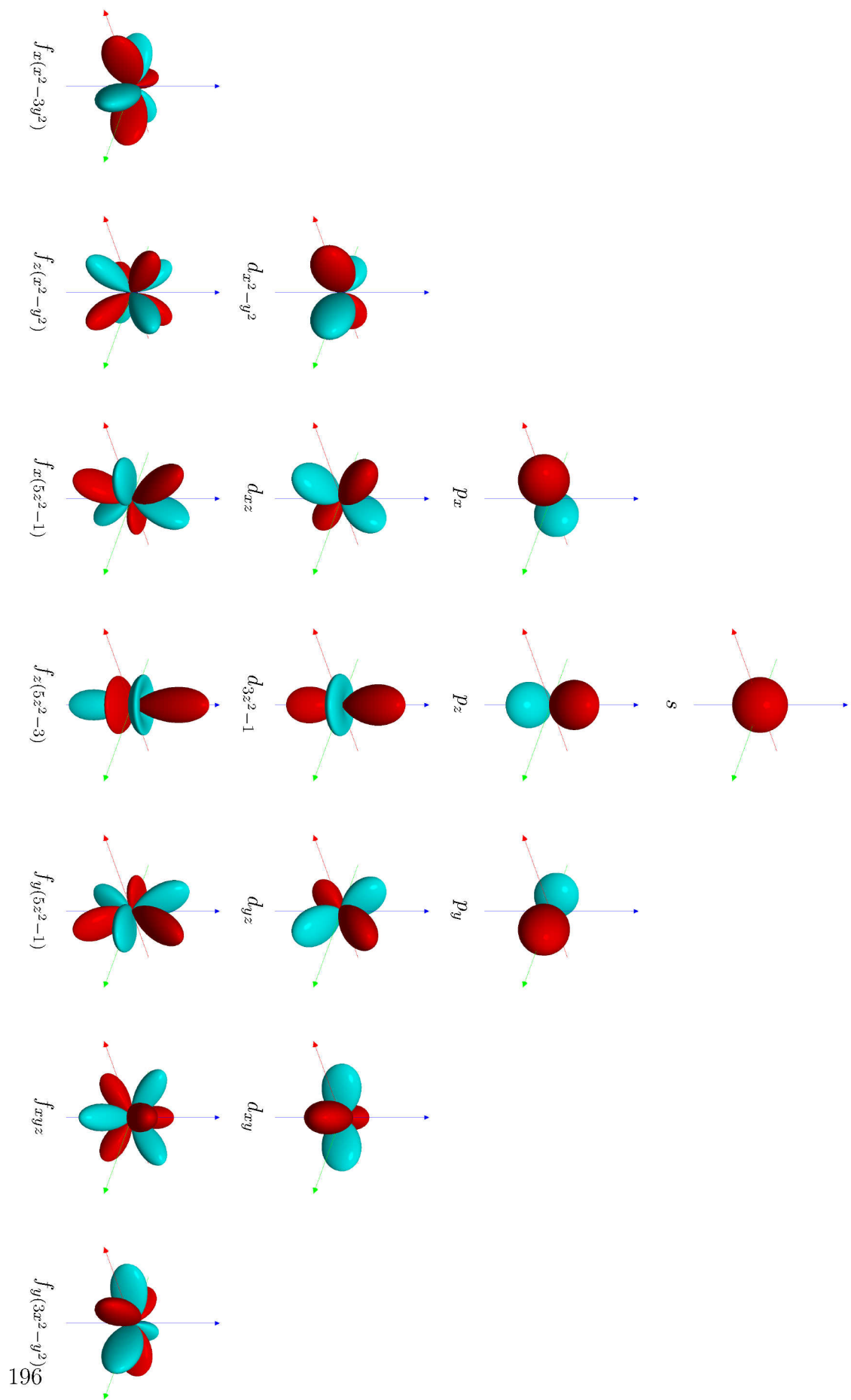


A.5. The first few cubic harmonics

Table A.5.: The first few cubic harmonics. x, y, z are the components of the unit vector $\hat{\mathbf{r}}$ (direction cosines).

s	$= Y_{0,0}$	$= \sqrt{\frac{1}{4\pi}}$
p_x	$= \frac{1}{\sqrt{2}}(Y_{1,-1} - Y_{1,1})$	$= \sqrt{\frac{3}{4\pi}} x$
p_z	$= Y_{1,0}$	$= \sqrt{\frac{3}{4\pi}} z$
p_y	$= \frac{i}{\sqrt{2}}(Y_{1,-1} + Y_{1,1})$	$= \sqrt{\frac{3}{4\pi}} y$
$d_{x^2-y^2}$	$= \frac{1}{\sqrt{2}}(Y_{2,-2} + Y_{2,2})$	$= \sqrt{\frac{15}{16\pi}} (x^2 - y^2)$
d_{xz}	$= \frac{1}{\sqrt{2}}(Y_{2,-1} - Y_{2,1})$	$= \sqrt{\frac{15}{4\pi}} xz$
d_{3z^2-1}	$= Y_{2,0}$	$= \sqrt{\frac{5}{16\pi}} (3z^2 - 1)$
d_{yz}	$= \frac{i}{\sqrt{2}}(Y_{2,-1} + Y_{2,1})$	$= \sqrt{\frac{15}{4\pi}} yz$
d_{xy}	$= \frac{i}{\sqrt{2}}(Y_{2,-2} - Y_{2,2})$	$= \sqrt{\frac{15}{4\pi}} xy$
$f_{x(x^2-3y^2)}$	$= \frac{1}{\sqrt{2}}(Y_{3,-3} - Y_{3,3})$	$= \sqrt{\frac{35}{32\pi}} x(x^2 - 3y^2)$
$f_{z(x^2-y^2)}$	$= \frac{1}{\sqrt{2}}(Y_{3,-2} + Y_{3,2})$	$= \sqrt{\frac{105}{16\pi}} z(x^2 - y^2)$
$f_{x(5z^2-1)}$	$= \frac{1}{\sqrt{2}}(Y_{3,-1} - Y_{3,1})$	$= \sqrt{\frac{21}{32\pi}} x(5z^2 - 1)$
$f_{z(5z^2-3)}$	$= Y_{3,0}$	$= \sqrt{\frac{7}{16\pi}} z(5z^2 - 3)$
$f_{y(5z^2-1)}$	$= \frac{i}{\sqrt{2}}(Y_{3,-1} + Y_{3,1})$	$= \sqrt{\frac{21}{32\pi}} y(5z^2 - 1)$
f_{xyz}	$= \frac{i}{\sqrt{2}}(Y_{3,-2} - Y_{3,2})$	$= \sqrt{\frac{105}{4\pi}} xyz$
$f_{y(3x^2-y^2)}$	$= \frac{i}{\sqrt{2}}(Y_{3,-3} + Y_{3,3})$	$= \sqrt{\frac{35}{32\pi}} y(3x^2 - y^2)$

Figure A.2.: The first few cubic harmonics.



A.6. Orbital overlap angular dependences

Table A.6.: Angular dependences of s -, p -, and d -orbital overlaps $A_{\alpha\beta}^{(\lambda)}(\hat{\mathbf{r}})$ (8.36). The unit vector $\hat{\mathbf{r}}$ points from orbital- α to orbital- β . x , y , z are the components of the unit vector $\hat{\mathbf{r}}$ (direction cosines). The cubic harmonics are sorted according to Table A.5. Note that $A_{\beta\alpha}^{(\lambda)}(\hat{\mathbf{r}}) = A_{\alpha\beta}^{(\lambda)}(-\hat{\mathbf{r}})$.

α	β	$A_{\alpha\beta}^{(0)}(\hat{\mathbf{r}})$	$A_{\alpha\beta}^{(1)}(\hat{\mathbf{r}})$	$A_{\alpha\beta}^{(2)}(\hat{\mathbf{r}})$	$A_{\alpha\beta}^{(3)}(\hat{\mathbf{r}})$	$A_{\alpha\beta}^{(4)}(\hat{\mathbf{r}})$
s	s	1				
s	p_x		$-\sqrt{3}x$			
s	p_z		$-\sqrt{3}z$			
s	p_y		$-\sqrt{3}y$			
s	$d_{x^2-y^2}$			$\frac{\sqrt{15}}{2}(x^2 - y^2)$		
s	d_{xz}			$\sqrt{15}xz$		
s	d_{3z^2-1}			$\frac{\sqrt{5}}{2}(3z^2 - 1)$		
s	d_{yz}			$\sqrt{15}yz$		
s	d_{xy}			$\sqrt{15}xy$		
p_x	p_x	1		$1 - 3x^2$		
p_x	p_z	0		$-3xz$		
p_x	p_y	0		$-3xy$		
p_x	$d_{x^2-y^2}$		$-\frac{3\sqrt{5}}{5}x$		$\frac{3\sqrt{5}}{10}x(5x^2 - 5y^2 - 2)$	
p_x	d_{xz}		$-\frac{3\sqrt{5}}{5}z$		$\frac{3\sqrt{5}}{5}z(5x^2 - 1)$	
p_x	d_{3z^2-1}		$\frac{\sqrt{15}}{5}x$		$\frac{3\sqrt{15}}{10}x(5z^2 - 1)$	
p_x	d_{yz}		0		$3\sqrt{5}xyz$	
p_x	d_{xy}		$-\frac{3\sqrt{5}}{5}y$		$\frac{3\sqrt{5}}{5}y(5x^2 - 1)$	
p_z	p_z	1		$1 - 3z^2$		
p_z	p_y	0		$-3yz$		
p_z	$d_{x^2-y^2}$		0		$\frac{3\sqrt{5}}{2}z(x^2 - y^2)$	
p_z	d_{xz}		$-\frac{3\sqrt{5}}{5}x$		$\frac{3\sqrt{5}}{5}x(5z^2 - 1)$	
p_z	d_{3z^2-1}		$-\frac{2\sqrt{15}}{5}z$		$\frac{3\sqrt{15}}{10}z(5z^2 - 3)$	
p_z	d_{yz}		$-\frac{3\sqrt{5}}{5}y$		$\frac{3\sqrt{5}}{5}y(5z^2 - 1)$	
p_z	d_{xy}		0		$3\sqrt{5}xyz$	

Table A.6 continued.

α	β	$A_{\alpha\beta}^{(0)}(\hat{\mathbf{r}})$	$A_{\alpha\beta}^{(1)}(\hat{\mathbf{r}})$	$A_{\alpha\beta}^{(2)}(\hat{\mathbf{r}})$	$A_{\alpha\beta}^{(3)}(\hat{\mathbf{r}})$	$A_{\alpha\beta}^{(4)}(\hat{\mathbf{r}})$
p_y	p_y	1		$1 - 3y^2$		
p_y	$d_{x^2-y^2}$		$\frac{3\sqrt{5}}{5}y$		$\frac{3\sqrt{5}}{10}y(5x^2 - 5y^2 + 2)$	
p_y	d_{xz}		0		$3\sqrt{5}xyz$	
p_y	d_{3z^2-1}		$\frac{\sqrt{15}}{5}y$		$\frac{3\sqrt{5}}{10}y(5z^2 - 1)$	
p_y	d_{yz}		$-\frac{3\sqrt{5}}{5}z$		$\frac{3\sqrt{5}}{5}z(5y^2 - 1)$	
p_y	d_{xy}		$-\frac{3\sqrt{5}}{5}x$		$\frac{3\sqrt{5}}{5}x(5y^2 - 1)$	
$d_{x^2-y^2}$	$d_{x^2-y^2}$	1		$\frac{5}{7}(3z^2 - 1)$		$\frac{3}{28}(35(x^2 - y^2)^2 + 20z^2 - 16)$
$d_{x^2-y^2}$	d_{xz}	0		$-\frac{15}{7}xz$		$\frac{15}{14}xz(7(x^2 - y^2) - 2)$
$d_{x^2-y^2}$	d_{3z^2-1}	0		$\frac{5\sqrt{3}}{7}(x^2 - y^2)$		$\frac{15\sqrt{3}}{28}(x^2 - y^2)(7z^2 - 1)$
$d_{x^2-y^2}$	d_{yz}	0		$\frac{15}{7}yz$		$\frac{15}{14}yz(7(x^2 - y^2) + 2)$
$d_{x^2-y^2}$	d_{xy}	0		0		$\frac{15}{2}xy(x^2 - y^2)$
d_{xz}	d_{xz}	1		$\frac{5}{7}(3y^2 - 1)$		$\frac{3}{7}(35x^2z^2 + 5y^2 - 4)$
d_{xz}	d_{3z^2-1}	0		$-\frac{5\sqrt{3}}{7}xz$		$\frac{15\sqrt{3}}{14}xz(7z^2 - 3)$
d_{xz}	d_{yz}	0		$-\frac{15}{7}xz$		$\frac{15}{7}xy(7z^2 - 1)$
d_{xz}	d_{xy}	0		$-\frac{15}{7}yz$		$\frac{15}{7}yz(7x^2 - 1)$
d_{3z^2-1}	d_{3z^2-1}	1		$-\frac{5}{7}(3z^2 - 1)$		$\frac{9}{28}(35z^4 - 30z^2 + 3)$
d_{3z^2-1}	d_{yz}	0		$-\frac{5\sqrt{3}}{7}yz$		$\frac{15\sqrt{3}}{14}yz(7z^2 - 3)$
d_{3z^2-1}	d_{xy}	0		$\frac{10\sqrt{3}}{7}xy$		$\frac{15\sqrt{3}}{14}xy(7z^2 - 1)$
d_{yz}	d_{yz}	1		$\frac{5}{7}(3x^2 - 1)$		$\frac{3}{7}(35y^2z^2 + 5x^2 - 4)$
d_{yz}	d_{xy}	0		$-\frac{15}{7}xz$		$\frac{15}{7}xz(7y^2 - 1)$
d_{xy}	d_{xy}	1		$\frac{5}{7}(3z^2 - 1)$		$\frac{3}{7}(35x^2y^2 + 5z^2 - 4)$

B. Multiplet reference tables

B.1. Slater-Condon and Racah parameters

References [59, 60, 61, 22, 62, 21].

Table B.1.: Slater-Condon and Racah parameters

Shell	$F_{(k)}$	Racah	
s	$F_{(0)} = F^{(0)}$	$A = F^{(0)}$	$= F_{(0)}$
p	$F_{(0)} = F^{(0)}$ $F_{(2)} = \frac{1}{25} F^{(2)}$	$A = F^{(0)}$ $B = \frac{1}{25} F^{(2)}$	$= F_{(0)}$ $= F_{(2)}$
d	$F_{(0)} = F^{(0)}$ $F_{(2)} = \frac{1}{49} F^{(2)}$ $F_{(4)} = \frac{1}{441} F^{(4)}$	$A = F^{(0)} - \frac{1}{9} F^{(4)}$ $B = \frac{1}{49} F^{(2)} - \frac{5}{441} F^{(4)}$ $C = \frac{5}{63} F^{(4)}$	$= F_{(0)} - 49 F_{(4)}$ $= F_{(2)} - 5 F_{(4)}$ $= 35 F_{(4)}$
f	$F_{(0)} = F^{(0)}$ $F_{(2)} = \frac{1}{225} F^{(2)}$ $F_{(4)} = \frac{1}{1089} F^{(4)}$ $F_{(6)} = \frac{25}{184041} F^{(6)}$	$A = F^{(0)} - \frac{7}{363} F^{(4)} - \frac{100}{1573} F^{(6)}$ $B = \frac{1}{225} F^{(2)} + \frac{2}{1815} F^{(4)} - \frac{35}{14157} F^{(6)}$ $C = \frac{7}{5445} F^{(4)} - \frac{70}{61347} F^{(6)}$ $D = \frac{350}{5577} F^{(6)}$	$= F_{(0)} - 21 F_{(4)} - 468 F_{(6)}$ $= F_{(2)} + \frac{6}{5} F_{(4)} - \frac{91}{5} F_{(6)}$ $= \frac{7}{5} F_{(4)} - \frac{42}{5} F_{(6)}$ $= 462 F_{(6)}$

B.2. Table of open-shell multiplets

Table B.2.: Atomic multiplets for open s , p , d and f shells.

Shell	Multiplets												
$s^0 s^2$	$1S$												
s^1	$2S$												
$p^0 p^6$	$1S$												
$p^1 p^5$		$2P$											
$p^2 p^4$	$1S$	$3P$	$1D$										
p^3	$4S$	$2P$	$2D$										
$d^0 d^{10}$	$1S$												
$d^1 d^9$			$2D$										
$d^2 d^8$	$1S$	$3P$	$1D$	$3F$	$1G$								
$d^3 d^7$		$2P^4P$	$2D$	$2F^4F$	$2G$	$2H$							
$d^4 d^6$	$1S$	$3P$	$1D^3D^5D$	$1F^3F$	$1G^3G$	$3H$	$1I$						
d^5	$2S^6S$	$2P^4P$	$2D^4D$	$2F^4F$	$2G^4G$	$2H$	$2I$						
$f^0 f^{14}$	$1S$												
$f^1 f^{13}$			$2F$										
$f^2 f^{12}$	$1S$	$3P$	$1D$	$3F$	$1G$	$3H$	$1I$						
$f^3 f^{11}$	$4S$	$2P$	$2D^4D$	$2F^4F$	$2G^4G$	$2H$	$2I^4I$	$2K$	$2L$				
$f^4 f^{10}$	$1S^5S$	$3P$	$1D^3D^5D$	$1F^3F^5F$	$1G^3G^5G$	$1H^3H$	$1I^3I^5I$	$1K^3K$	$1L^3L$	$3M$			
$f^5 f^9$	$4S$	$2P^4P^6P$	$2D^4D$	$2F^4F^6F$	$2G^4G$	$2H^4H^6H$	$2I^4I$	$2K^4K$	$2L^4L$	$2M^4M$			
$f^6 f^8$	$1S^5S$	$1P^3P^5P$	$1D^3D^5D$	$1F^3F^5F^7F$	$1G^3G^5G$	$1H^3H^5H$	$1I^3I^5I$	$1K^3K^5K$	$1L^3L^5L$	$2N$			
f^7	$2S^4S^8S$	$2P^4P^6P$	$2D^4D^6D$	$2F^4F^6F$	$2G^4G^6G$	$2H^4H^6H$	$2I^4I^6I$	$2K^4K$	$2L^4L$	$2M^4M$			
										$2N^4N$			
										$2O$			
										$1Q$			
										$2Q$			

B.3. Hund's rule ground state term energies

Table B.3.: Hund's rule ground state term energies

s^0	1S	0
s^1	2S	0
s^2	1S	$F^{(0)}$
p^0	1S	0
p^1	2P	0
p^2	3P	$F^{(0)} - \frac{1}{5}F^{(2)}$
p^3	4S	$3F^{(0)} - \frac{3}{5}F^{(2)}$
p^4	3P	$6F^{(0)} - \frac{3}{5}F^{(2)}$
p^5	2P	$10F^{(0)} - \frac{4}{5}F^{(2)}$
p^6	1S	$15F^{(0)} - \frac{6}{5}F^{(2)}$
d^0	1S	0
d^1	2D	0
d^2	3F	$F^{(0)} - \frac{8}{49}F^{(2)} - \frac{1}{49}F^{(4)}$
d^3	4F	$3F^{(0)} - \frac{15}{49}F^{(2)} - \frac{8}{49}F^{(4)}$
d^4	5D	$6F^{(0)} - \frac{3}{7}F^{(2)} - \frac{3}{7}F^{(4)}$
d^5	6S	$10F^{(0)} - \frac{5}{7}F^{(2)} - \frac{5}{7}F^{(4)}$
d^6	5D	$15F^{(0)} - \frac{5}{7}F^{(2)} - \frac{5}{7}F^{(4)}$
d^7	4F	$21F^{(0)} - \frac{43}{49}F^{(2)} - \frac{36}{49}F^{(4)}$
d^8	3F	$28F^{(0)} - \frac{50}{49}F^{(2)} - \frac{43}{49}F^{(4)}$
d^9	2D	$36F^{(0)} - \frac{8}{7}F^{(2)} - \frac{8}{7}F^{(4)}$
d^{10}	1S	$45F^{(0)} - \frac{10}{7}F^{(2)} - \frac{10}{7}F^{(4)}$

f^0	1S	0
f^1	2F	0
f^2	3H	$F^{(0)} - \frac{1}{9}F^{(2)} - \frac{17}{363}F^{(4)} - \frac{25}{14157}F^{(6)}$
f^3	4I	$3F^{(0)} - \frac{13}{45}F^{(2)} - \frac{363}{80}F^{(4)} - \frac{14157}{2075}F^{(6)}$
f^4	5I	$6F^{(0)} - \frac{19}{45}F^{(2)} - \frac{363}{116}F^{(4)} - \frac{14157}{4975}F^{(6)}$
f^5	6H	$10F^{(0)} - \frac{23}{45}F^{(2)} - \frac{116}{363}F^{(4)} - \frac{14157}{14157}F^{(6)}$
f^6	7F	$15F^{(0)} - \frac{2}{3}F^{(2)} - \frac{5}{11}F^{(4)} - \frac{250}{429}F^{(6)}$
f^7	8S	$21F^{(0)} - \frac{14}{15}F^{(2)} - \frac{11}{7}F^{(4)} - \frac{350}{429}F^{(6)}$
f^8	7F	$28F^{(0)} - \frac{14}{15}F^{(2)} - \frac{11}{7}F^{(4)} - \frac{350}{429}F^{(6)}$
f^9	6H	$36F^{(0)} - \frac{47}{45}F^{(2)} - \frac{248}{363}F^{(4)} - \frac{11575}{14157}F^{(6)}$
f^{10}	5I	$45F^{(0)} - \frac{11}{9}F^{(2)} - \frac{278}{363}F^{(4)} - \frac{11975}{14157}F^{(6)}$
f^{11}	4I	$55F^{(0)} - \frac{61}{45}F^{(2)} - \frac{311}{363}F^{(4)} - \frac{13625}{14157}F^{(6)}$
f^{12}	3H	$66F^{(0)} - \frac{13}{9}F^{(2)} - \frac{347}{363}F^{(4)} - \frac{16525}{14157}F^{(6)}$
f^{13}	2F	$78F^{(0)} - \frac{8}{5}F^{(2)} - \frac{12}{11}F^{(4)} - \frac{200}{143}F^{(6)}$
f^{14}	1S	$91F^{(0)} - \frac{28}{15}F^{(2)} - \frac{14}{11}F^{(4)} - \frac{700}{429}F^{(6)}$

B.4. Multiplet term energies

Table B.4.: Multiplet term energies for open s , p , d and f shells.

p^0	0	1S	$15F^{(0)} - \frac{6}{5}F^{(2)}$	p^6										
p^1	0	2P	$10F^{(0)} - \frac{4}{5}F^{(2)}$	p^5										
p^2	$F^{(0)} - \frac{1}{5}F^{(2)}$ $F^{(0)} + \frac{1}{25}F^{(2)}$ $F^{(0)} + \frac{2}{5}F^{(2)}$	3P 1D 1S	$6F^{(0)} - \frac{3}{5}F^{(2)}$ $6F^{(0)} - \frac{9}{25}F^{(2)}$ $6F^{(0)}$	p^4										
p^3	$3F^{(0)} - \frac{3}{5}F^{(2)}$ $3F^{(0)} - \frac{6}{25}F^{(2)}$ $3F^{(0)}$	4S 2D 2P	<table> <tr> <td>s^0</td><td>0</td><td>1S</td><td>$F^{(0)}$</td><td>s^2</td></tr> <tr> <td>s^1</td><td>0</td><td>2S</td><td></td><td></td></tr> </table>		s^0	0	1S	$F^{(0)}$	s^2	s^1	0	2S		
s^0	0	1S	$F^{(0)}$	s^2										
s^1	0	2S												

d^0	0	1S	$45F^{(0)} - \frac{10}{7}F^{(2)} - \frac{10}{7}F^{(4)}$	d^{10}
d^1	0	2D	$36F^{(0)} - \frac{8}{7}F^{(2)} - \frac{8}{7}F^{(4)}$	d^9
d^2	$F^{(0)} - \frac{8}{49}F^{(2)} - \frac{1}{49}F^{(4)}$ $F^{(0)} + \frac{1}{7}F^{(2)} - \frac{4}{21}F^{(4)}$ $F^{(0)} + \frac{4}{49}F^{(2)} + \frac{1}{441}F^{(4)}$ $F^{(0)} - \frac{3}{49}F^{(2)} + \frac{4}{49}F^{(4)}$ $F^{(0)} + \frac{2}{7}F^{(2)} + \frac{2}{7}F^{(4)}$	3F 3P 1G 1D 1S	$28F^{(0)} - \frac{50}{49}F^{(2)} - \frac{43}{49}F^{(4)}$ $28F^{(0)} - \frac{5}{7}F^{(2)} - \frac{22}{21}F^{(4)}$ $28F^{(0)} - \frac{38}{49}F^{(2)} - \frac{377}{441}F^{(4)}$ $28F^{(0)} - \frac{45}{49}F^{(2)} - \frac{38}{49}F^{(4)}$ $28F^{(0)} - \frac{4}{7}F^{(2)} - \frac{4}{7}F^{(4)}$	d^8
d^3	$3F^{(0)} - \frac{15}{49}F^{(2)} - \frac{8}{49}F^{(4)}$ $3F^{(0)} - \frac{1}{3}F^{(4)}$ $3F^{(0)} - \frac{6}{49}F^{(2)} - \frac{4}{147}F^{(4)}$ $3F^{(0)} - \frac{11}{49}F^{(2)} + \frac{13}{441}F^{(4)}$ $3F^{(0)} + \frac{9}{49}F^{(2)} - \frac{29}{147}F^{(4)}$ $3F^{(0)} + \frac{5}{49}F^{(2)} + \frac{1}{147}F^{(4)} - \Delta$ $3F^{(0)} + \frac{5}{49}F^{(2)} + \frac{1}{147}F^{(4)} + \Delta$ $3F^{(0)} - \frac{6}{49}F^{(2)} - \frac{4}{147}F^{(4)}$	4F 4P 2H 2G 2F 2D 2D 2P	$21F^{(0)} - \frac{43}{49}F^{(2)} - \frac{36}{49}F^{(4)}$ $21F^{(0)} - \frac{4}{7}F^{(2)} - \frac{19}{21}F^{(4)}$ $21F^{(0)} - \frac{34}{49}F^{(2)} - \frac{88}{147}F^{(4)}$ $21F^{(0)} - \frac{39}{49}F^{(2)} - \frac{239}{441}F^{(4)}$ $21F^{(0)} - \frac{19}{49}F^{(2)} - \frac{113}{147}F^{(4)}$ $21F^{(0)} - \frac{23}{49}F^{(2)} - \frac{83}{147}F^{(4)} - \Delta$ $21F^{(0)} - \frac{23}{49}F^{(2)} - \frac{83}{147}F^{(4)} + \Delta$ $21F^{(0)} - \frac{34}{49}F^{(2)} - \frac{88}{147}F^{(4)}$	d^7
$\Delta = \frac{1}{147}\sqrt{1737F^{(2)2} - 1650F^{(2)}F^{(4)} + 925F^{(4)2}}$				

d^4	$6F^{(0)} - \frac{3}{7}F^{(2)} - \frac{3}{7}F^{(4)}$	5D	$15F^{(0)} - \frac{5}{7}F^{(2)} - \frac{5}{7}F^{(4)}$	d^6
	$6F^{(0)} - \frac{17}{49}F^{(2)} - \frac{23}{147}F^{(4)}$	3H	$15F^{(0)} - \frac{31}{49}F^{(2)} - \frac{65}{147}F^{(4)}$	
	$6F^{(0)} - \frac{12}{49}F^{(2)} - \frac{94}{441}F^{(4)}$	3G	$15F^{(0)} - \frac{26}{49}F^{(2)} - \frac{220}{441}F^{(4)}$	
	$6F^{(0)} - \frac{5}{49}F^{(2)} - \frac{17}{98}F^{(4)} - \Delta_1$	3F	$15F^{(0)} - \frac{19}{49}F^{(2)} - \frac{45}{98}F^{(4)} - \Delta_1$	
	$6F^{(0)} - \frac{5}{49}F^{(2)} - \frac{17}{98}F^{(4)} + \Delta_1$	3F	$15F^{(0)} - \frac{19}{49}F^{(2)} - \frac{45}{98}F^{(4)} + \Delta_1$	
	$6F^{(0)} - \frac{5}{49}F^{(2)} - \frac{43}{147}F^{(4)}$	3D	$15F^{(0)} - \frac{19}{49}F^{(2)} - \frac{85}{147}F^{(4)}$	
	$6F^{(0)} - \frac{5}{49}F^{(2)} - \frac{17}{98}F^{(4)} - \Delta_2$	3P	$15F^{(0)} - \frac{19}{49}F^{(2)} - \frac{45}{98}F^{(4)} - \Delta_2$	
	$6F^{(0)} - \frac{5}{49}F^{(2)} - \frac{17}{98}F^{(4)} + \Delta_2$	3P	$15F^{(0)} - \frac{19}{49}F^{(2)} - \frac{45}{98}F^{(4)} + \Delta_2$	
	$6F^{(0)} - \frac{15}{49}F^{(2)} - \frac{1}{49}F^{(4)}$	1I	$15F^{(0)} - \frac{29}{49}F^{(2)} - \frac{15}{49}F^{(4)}$	
	$6F^{(0)} - \frac{5}{49}F^{(2)} - \frac{13}{882}F^{(4)} - \Delta_3$	1G	$15F^{(0)} - \frac{19}{49}F^{(2)} - \frac{265}{882}F^{(4)} - \Delta_3$	
	$6F^{(0)} - \frac{5}{49}F^{(2)} - \frac{13}{882}F^{(4)} + \Delta_3$	1G	$15F^{(0)} - \frac{19}{49}F^{(2)} - \frac{265}{882}F^{(4)} + \Delta_3$	
	$6F^{(0)} - \frac{4}{21}F^{(4)}$	1F	$15F^{(0)} - \frac{2}{7}F^{(2)} - \frac{10}{21}F^{(4)}$	
	$6F^{(0)} + \frac{9}{49}F^{(2)} - \frac{17}{98}F^{(4)} - \Delta_4$	1D	$15F^{(0)} - \frac{5}{49}F^{(2)} - \frac{45}{98}F^{(4)} - \Delta_4$	
	$6F^{(0)} + \frac{9}{49}F^{(2)} - \frac{17}{98}F^{(4)} + \Delta_4$	1D	$15F^{(0)} - \frac{5}{49}F^{(2)} - \frac{45}{98}F^{(4)} + \Delta_4$	
	$6F^{(0)} + \frac{10}{49}F^{(2)} + \frac{2}{147}F^{(4)} - \Delta_5$	1S	$15F^{(0)} - \frac{4}{49}F^{(2)} - \frac{40}{147}F^{(4)} - \Delta_5$	
	$6F^{(0)} + \frac{10}{49}F^{(2)} + \frac{2}{147}F^{(4)} + \Delta_5$	1S	$15F^{(0)} - \frac{4}{49}F^{(2)} - \frac{40}{147}F^{(4)} + \Delta_5$	
$\Delta_1 = \frac{1}{294} \sqrt{5508F^{(2)2} - 4860F^{(2)}F^{(4)} + 2225F^{(4)2}}$				
$\Delta_2 = \frac{1}{294} \sqrt{8208F^{(2)2} - 9960F^{(2)}F^{(4)} + 4225F^{(4)2}}$				
$\Delta_3 = \frac{1}{294} \sqrt{6372F^{(2)2} - 7500F^{(2)}F^{(4)} + 3425F^{(4)2}}$				
$\Delta_4 = \frac{1}{294} \sqrt{11664F^{(2)2} - 10440F^{(2)}F^{(4)} + 3425F^{(4)2}}$				
$\Delta_5 = \frac{2}{147} \sqrt{1737F^{(2)2} - 1650F^{(2)}F^{(4)} + 925F^{(4)2}}$				

B. Multiplet reference tables

d^5	$10F^{(0)} - \frac{5}{7}F^{(2)} - \frac{5}{7}F^{(4)}$	6S
	$10F^{(0)} - \frac{25}{49}F^{(2)} - \frac{190}{441}F^{(4)}$	4G
	$10F^{(0)} - \frac{13}{49}F^{(2)} - \frac{20}{49}F^{(4)}$	4F
	$10F^{(0)} - \frac{18}{49}F^{(2)} - \frac{25}{49}F^{(4)}$	4D
	$10F^{(0)} - \frac{4}{7}F^{(2)} - \frac{5}{21}F^{(4)}$	4P
	$10F^{(0)} - \frac{24}{49}F^{(2)} - \frac{10}{49}F^{(4)}$	2I
	$10F^{(0)} - \frac{22}{49}F^{(2)} - \frac{10}{147}F^{(4)}$	2H
	$10F^{(0)} + \frac{3}{49}F^{(2)} - \frac{155}{441}F^{(4)}$	2G
	$10F^{(0)} - \frac{13}{49}F^{(2)} - \frac{145}{441}F^{(4)}$	2G
	$10F^{(0)} - \frac{9}{49}F^{(2)} - \frac{55}{147}F^{(4)}$	2F
	$10F^{(0)} - \frac{25}{49}F^{(2)} - \frac{5}{147}F^{(4)}$	2F
	$10F^{(0)} - \frac{4}{49}F^{(2)} - \frac{40}{147}F^{(4)}$	2D
	$10F^{(0)} - \frac{3}{49}F^{(2)} - \frac{10}{49}F^{(4)} - \Delta$	2D
	$10F^{(0)} - \frac{3}{49}F^{(2)} - \frac{10}{49}F^{(4)} + \Delta$	2D
	$10F^{(0)} + \frac{20}{49}F^{(2)} - \frac{80}{147}F^{(4)}$	2P
	$10F^{(0)} - \frac{3}{49}F^{(2)} - \frac{65}{147}F^{(4)}$	2S
$\Delta = \frac{1}{147}\sqrt{4617F^{(2)2} + 540F^{(2)}F^{(4)} - 500F^{(4)2}}$		

f^0	0	1S	$91F^{(0)} - \frac{28}{15}F^{(2)} - \frac{14}{11}F^{(4)} - \frac{700}{429}F^{(6)}$	f^{14}
f^1	0	2F	$78F^{(0)} - \frac{8}{5}F^{(2)} - \frac{12}{11}F^{(4)} - \frac{200}{143}F^{(6)}$	f^{13}
f^2	$F^{(0)} - \frac{1}{9}F^{(2)} - \frac{17}{363}F^{(4)} - \frac{25}{14157}F^{(6)}$	3H	$66F^{(0)} - \frac{13}{9}F^{(2)} - \frac{347}{363}F^{(4)} - \frac{16525}{14157}F^{(6)}$	f^{12}
	$F^{(0)} - \frac{2}{45}F^{(2)} - \frac{1}{33}F^{(4)} - \frac{50}{1287}F^{(6)}$	3F	$66F^{(0)} - \frac{62}{45}F^{(2)} - \frac{31}{33}F^{(4)} - \frac{1550}{1287}F^{(6)}$	
	$F^{(0)} + \frac{1}{5}F^{(2)} + \frac{1}{33}F^{(4)} - \frac{25}{143}F^{(6)}$	3P	$66F^{(0)} - \frac{17}{15}F^{(2)} - \frac{29}{33}F^{(4)} - \frac{575}{429}F^{(6)}$	
	$F^{(0)} + \frac{1}{9}F^{(2)} + \frac{1}{121}F^{(4)} + \frac{25}{184041}F^{(6)}$	1I	$66F^{(0)} - \frac{11}{9}F^{(2)} - \frac{109}{121}F^{(4)} - \frac{214475}{184041}F^{(6)}$	
	$F^{(0)} - \frac{2}{15}F^{(2)} + \frac{97}{1089}F^{(4)} + \frac{50}{4719}F^{(6)}$	1G	$66F^{(0)} - \frac{22}{15}F^{(2)} - \frac{893}{1089}F^{(4)} - \frac{5450}{4719}F^{(6)}$	
	$F^{(0)} + \frac{19}{225}F^{(2)} - \frac{1}{11}F^{(4)} + \frac{125}{1287}F^{(6)}$	1D	$66F^{(0)} - \frac{281}{225}F^{(2)} - F^{(4)} - \frac{125}{117}F^{(6)}$	
	$F^{(0)} + \frac{4}{15}F^{(2)} + \frac{2}{11}F^{(4)} + \frac{100}{429}F^{(6)}$	1S	$66F^{(0)} - \frac{16}{15}F^{(2)} - \frac{8}{11}F^{(4)} - \frac{400}{429}F^{(6)}$	

B.5. Multiplet and spin-orbit spectral variances

Table B.5.: Multiplet and spin-orbit spectral variances for open s , p , d and f shells.

Shell	H_U	H_{SO}
$p^1 p^5$		$\frac{1}{2} \Xi^2$
$p^2 p^4$	$\frac{18}{625} (F^{(2)})^2$	$\frac{4}{5} \Xi^2$
p^3	$\frac{27}{625} (F^{(2)})^2$	$\frac{9}{10} \Xi^2$
$d^1 d^9$		$\frac{3}{2} \Xi^2$
$d^2 d^8$	$\frac{530}{27783} (F^{(2)})^2 - \frac{200}{27783} F^{(2)} F^{(4)} + \frac{250}{27783} (F^{(4)})^2$	$\frac{8}{3} \Xi^2$
$d^3 d^7$	$\frac{265}{6174} (F^{(2)})^2 - \frac{50}{3087} F^{(2)} F^{(4)} + \frac{125}{6174} (F^{(4)})^2$	$\frac{7}{2} \Xi^2$
$d^4 d^6$	$\frac{1325}{21609} (F^{(2)})^2 - \frac{500}{21609} F^{(2)} F^{(4)} + \frac{625}{21609} (F^{(4)})^2$	$4 \Xi^2$
d^5	$\frac{13250}{194481} (F^{(2)})^2 - \frac{5000}{194481} F^{(2)} F^{(4)} + \frac{6250}{194481} (F^{(4)})^2$	$\frac{25}{6} \Xi^2$
$f^1 f^{13}$		$3 \Xi^2$
$f^2 f^{12}$	$\frac{7508}{570375} (F^{(2)})^2 + \frac{8}{2535} F^{(2)} F^{(4)} + \frac{5090}{2024451} (F^{(4)})^2 - \frac{280}{59319} F^{(2)} F^{(6)} - \frac{7000}{8772621} F^{(4)} F^{(6)} + \frac{4287500}{1026396657} (F^{(6)})^2$	$\frac{72}{13} \Xi^2$
$f^3 f^{11}$	$\frac{3754}{114075} (F^{(2)})^2 + \frac{4}{507} F^{(2)} F^{(4)} + \frac{12725}{2024451} (F^{(4)})^2 - \frac{700}{59319} F^{(2)} F^{(6)} - \frac{17500}{8772621} F^{(4)} F^{(6)} + \frac{10718750}{1026396657} (F^{(6)})^2$	$\frac{99}{13} \Xi^2$
$f^4 f^{10}$	$\frac{7508}{139425} (F^{(2)})^2 + \frac{24}{1859} F^{(2)} F^{(4)} + \frac{25450}{2474329} (F^{(4)})^2 - \frac{1400}{72501} F^{(2)} F^{(6)} - \frac{105000}{32166277} F^{(4)} F^{(6)} + \frac{21437500}{1254484803} (F^{(6)})^2$	$\frac{120}{13} \Xi^2$
$f^5 f^9$	$\frac{30032}{418275} (F^{(2)})^2 + \frac{32}{1859} F^{(2)} F^{(4)} + \frac{101800}{7422987} (F^{(4)})^2 - \frac{5600}{217503} F^{(2)} F^{(6)} - \frac{140000}{32166277} F^{(4)} F^{(6)} + \frac{85750000}{3763454409} (F^{(6)})^2$	$\frac{135}{13} \Xi^2$
$f^6 f^8$	$\frac{105112}{1254825} (F^{(2)})^2 + \frac{112}{5577} F^{(2)} F^{(4)} + \frac{356300}{22268961} (F^{(4)})^2 - \frac{19600}{652509} F^{(2)} F^{(6)} - \frac{490000}{96498831} F^{(4)} F^{(6)} + \frac{300125000}{11290363227} (F^{(6)})^2$	$\frac{144}{13} \Xi^2$
f^7	$\frac{183946}{2091375} (F^{(2)})^2 + \frac{196}{9295} F^{(2)} F^{(4)} + \frac{124705}{7422987} (F^{(4)})^2 - \frac{6860}{217503} F^{(2)} F^{(6)} - \frac{171500}{32166277} F^{(4)} F^{(6)} + \frac{105043750}{3763454409} (F^{(6)})^2$	$\frac{147}{13} \Xi^2$

Acknowledgments

I am heartily thankful to my supervisors, Prof. Dr. Erik Koch and Prof. Dr. Carsten Honerkamp, for their guidance, encouragement, and support throughout my doctoral studies.

Prof. Koch has been always patient and willing to give me advices. We enjoy discussing problems and working out the solution together in a systematic way. It is always impressive to see that Prof. Koch thinks on a problem carefully and quickly converges to an elegant solution (ok, I also contribute, sometimes): an example was that once I was asking him about writing a simple program to solve a partition problem. It was the first time that he thought on this problem. He took out a pencil and a piece of paper, with a bit of thinking, wrote down a very elegant recursive pseudocode in a few minutes. I quickly implemented the algorithm in Python, and it worked out like magic. There are many similar examples like this, but perhaps the most remarkable aspect is that Prof. Koch cares about students. He pays attention to everyone in the group. He worries about if someone has problems or difficulties in their life. He cares about students in a very personal way, he is a wonderful man. Prof. Koch and I are together responsible for two master courses (I am the teaching assistant), Applied Quantum Mechanics and Correlated Electrons. He spends efforts to prepare the course materials carefully, and he often comes to me to ask about the students' feedbacks and how we can possibly improve the lectures. He is a very competent educator.

I would like to thank Prof. Dr. Carsten Honerkamp for being the second supervisor and for his helpful comments and discussions. I would like to thank my dear colleagues, Dr. Hermann Ulm, Dr. Hunter Sims, Dr. Khaldoon Ghanem, Michael Baumgärtel, Hoai Le Thi, Cica Gustiani, and Rodrigo Canales. I also would like to thank Prof. Dr. Eva Pavarini and the dear group members, Dr. Guoren Zhang, Dr. Amin Kiani, Dr. Esmaeel Sarvestani, Dr. Xuejing Zhang, Neda Samani, Julian Mußhoff, and Johannes Mitscherling. Certainly, I also thank the two master students Uriel Allan Aceves Rodriguez and Ana Maria Montero Martinez, for inviting me to join the everyday Aachen-Jülich bus trip. They are all super nice and helpful. Guoren is a very kind man, he once invited me to his place to celebrate the Chinese spring festival. He even gave me nice furnitures when he was about to leave Germany. Julian is a gentleman with always a friendly smile. He is a German student and we entered the research training group at the same time. I love talking with Julian, he is so kind. And he helped me a lot to understand more about Germany. I remember once my curtain collapsed and I had to find a saw to fix it. But who on earth would have a saw at home? I figured Julian must have one. I went to his place, and you know what happened, he provided me four different kinds of saws, saying, one is for cutting metal, the other three are for wood; among the three, one has finer

B. Multiplet reference tables

teeth, one has coarser teeth, and one is electric. I was completely impressed that he had the entire toolbox to build a house. I would say, Julian is very German.

Special thanks to my dear friend Khaldoon. Khaldoon is one of my seniors. Normally all the practical problems that I encountered, he would have encountered them one year earlier. He gave me very useful advices which greatly helped me and dramatically reduced the potential stupid mistakes that I might have otherwise made. Khaldoon and I were in the same master program, and he was the only one that I know who has received the top grades among all the subjects, which is amazing. One can easily realize his intellectual abilities when discussing a problem with him. He can quickly discover where the key of a problem is, summarize the pattern of the problem, and provide an elegant solution.

I want to thank all the students from the RWTH Aachen Simulation Sciences master program. It was a great pleasure to teach them in the exercise classes. I must admit that during the doctoral studies, I often got stuck on certain problems, ran out of ideas, and easily became frustrated. I even had a guilty feeling that I was wasting the money of the institute. However, standing in front of the students I felt that I was recharged. They were so enthusiastic to learn new things. They asked very good questions and sometimes came up with brilliant ideas to solve problems. The biggest encouragement was the moment that they told me that I was a good teacher. All those details pushed me forward along my doctoral period.

Finally, I would like to thank the Research Training Group (RTG1995) for funding the first part of my work, Jülich Supercomputing Centre (JSC) and Institute for Advanced Simulations (IAS) for funding the second part. I should also thank all of our RTG professors, students, and administration colleagues. Over the past few years, the RTG organized outstanding seminars and workshops, in which active discussions took place and fruitful results were presented. We RTG students presented a wide range of topics in the seminars, which introduced me to many different modern approaches in quantum many-body problems. I am thankful to all the activities and discussion which definitely broadened my views and enriched my understanding and knowledge in my scientific career.

Bibliography

- [1] E. Koch. Exchange mechanisms. In E. Pavarini, E. Koch, F. Anders, and M. Jarrell, editors, *Correlated Electrons: From Models to Materials*, volume 2 of *Reihe Modeling and Simulation*. Forschungszentrum Jülich, 2012. Available: <http://hdl.handle.net/2128/4611>.
- [2] E. P. Wigner and F. Seitz. Qualitative analysis of the cohesion in metals. In F. Seitz and D. Turnbull, editors, *Solid State Physics*, volume 1. Academic Press, 1955.
- [3] H. J. Lipkin. *Quantum Mechanics: New Approaches to Selected Topics*. Dover Publications, 2007.
- [4] E. Pavarini and E. Koch. Introduction. In E. Pavarini, E. Koch, and U. Schollwöck, editors, *Emergent Phenomena in Correlated Matter*, volume 3 of *Reihe Modeling and Simulation*. Forschungszentrum Jülich, 2013. Available: <http://hdl.handle.net/2128/5389>.
- [5] E. Koch. Many-electron states. In E. Pavarini, E. Koch, and U. Schollwöck, editors, *Emergent Phenomena in Correlated Matter*, volume 3 of *Reihe Modeling and Simulation*. Forschungszentrum Jülich, 2013. Available: <http://hdl.handle.net/2128/5389>.
- [6] A. Szabo and N. S. Ostlund. *Modern quantum chemistry: introduction to advanced electronic structure theory*. Dover Publications, 1996.
- [7] E. Koch. The Lanczos method. In E. Pavarini, E. Koch, A. Lichtenstein, and D. Vollhardt, editors, *The LDA+DMFT approach to strongly correlated materials*, volume 1 of *Reihe Modeling and Simulation*. Forschungszentrum Jülich, 2011. Available: <http://hdl.handle.net/2128/4467>.
- [8] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov. Electric field effect in atomically thin carbon films. *Science*, 306:666, 2004.
- [9] D. R. Cooper, B. D’Anjou, N. Ghattamaneni, B. Harack, M. Hilke, A. Horth, N. Majlis, M. Massicotte, L. Vandsburger, E. Whiteway, and V. Yu. Experimental review of graphene. *ISRN Condensed Matter Physics*, 2012, 2012.
- [10] Y. Obeng and P. Srinivasan. Graphene: Is it the future for semiconductors? an overview of the material, devices, and applications. *The Electrochemical Society Interface*, 20:47, 2011.

Bibliography

- [11] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864, 1964.
- [12] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev. A*, 140:1133, 1965.
- [13] S. Kotochigova, Z. H. Levine, E. L. Shirley, M. D. Stiles, and C. W. Clark. Atomic reference data for electronic structure calculations. National Institute of Standards and Technology, Gaithersburg, MD., 2009. doi:10.18434/T4ZP4F.
- [14] R. M. Martin. *Electronic Structure*. Cambridge University Press, 2004.
- [15] Q. Zhang. Calculations of atomic multiplets across the periodic table. Master’s thesis, RWTH Aachen, 2014. Available: <http://hdl.handle.net/2128/8437>.
- [16] D. J. Griffiths. *Introduction to Quantum Mechanics*. Pearson Prentice Hall, 2nd edition, 2005.
- [17] D. J. Griffiths. *Introduction to Electrodynamics*. Prentice Hall, 3rd edition, 1999.
- [18] P. Ziesche, S. Kurth, and J. P. Perdew. Density functionals from LDA to GGA. *Comput. Mater. Sci.*, 11:122, 1998.
- [19] J. M. MacLaren, D. P. Clougherty, M. E. McHenry, and M. M. Donovan. Parameterised local spin density exchange-correlation energies and potentials for electronic structure calculations i. zero temperature formalism. *Comput. Phys. Commun.*, 66:383, 1991.
- [20] R. Eder. Multiplets in transition metal ions. In E. Pavarini, E. Koch, F. Anders, and M. Jarrell, editors, *Correlated Electrons: From Models to Materials*, volume 2 of *Reihe Modeling and Simulation*. Forschungszentrum Jülich, 2012. Available: <http://hdl.handle.net/2128/4611>.
- [21] M. Weissbluth. *Atoms and Molecules*. Academic Press, 1978.
- [22] G. Racah. Theory of complex spectra. III. *Phys. Rev.*, 63:367, 1943.
- [23] C. W. Haigh. The theory of atomic spectroscopy: jj coupling, intermediate coupling, and configuration interaction. *J. Chem. Educ.*, 72:206, 1995.
- [24] A. Kramida, Y. Ralchenko, J. Reader, and NIST ASD Team. NIST atomic spectra database. National Institute of Standards and Technology, Gaithersburg, MD., 2018. doi:10.18434/T4W30F.
- [25] H. J. Zeiger and G. W. Pratt. *Magnetic interactions in solids*. Clarendon Press, 1973.
- [26] W. C. Martin, A. Musgrove, S. Kotochigova, and J. E. Sansonetti. Ground levels and ionization energies for the neutral atoms. National Institute of Standards and Technology, Gaithersburg, MD., 2013. doi:10.18434/T42P4C.

- [27] S. A. Goudsmit and P. I. Richards. The order of electron shells in ionized atoms. *Proc. Natl. Acad. Sci. U.S.A.*, 51:664, 1964.
- [28] D. P. Wong. Theoretical justification of Madelung’s rule. *J. Chem. Educ.*, 56:714, 1979.
- [29] C. Janet. *Classification hélicoidale des éléments chimiques*. Imprimerie départementale de l’Oise, 1928.
- [30] E. Scerri. The role of triads in the evolution of the periodic table: past and present. *J. Chem. Educ.*, 85:585, 2008.
- [31] P. J. Stewart. Charles Janet: unrecognized genius of the periodic system. *Found. Chem.*, 12:5, 2010.
- [32] E. Fluck. New notations in the periodic table. *Pure Appl. Chem.*, 60:431, 2009.
- [33] E. Scerri. Which elements belong in group 3 of the periodic table? *Chem. Int.*, 38:22, 2016.
- [34] D. C. Hoffman, D. M. Lee, and V. Pershina. Transactinide elements and future elements. In L. R. Morss, N. M. Edelstein, and J. Fuger, editors, *The Chemistry of the Actinide and Transactinide Elements*, page 1652. Springer Netherlands, 3rd edition, 2008.
- [35] J. C. Slater. *Quantum Theory of Atomic Structure*, volume 1. McGraw-Hill Book Company, 1960.
- [36] E. U. Condon and G. H. Shortley. *The theory of atomic spectra*. Cambridge University Press, 1951.
- [37] F. M. F. de Groot, J. C. Fuggle, B. T. Thole, and G. A. Sawatzky. 2p x-ray absorption of 3d transition-metal compounds: An atomic multiplet description including the crystal field. *Phys. Rev. B*, 42:5459, 1990.
- [38] B. T. Thole, G. van der Laan, J. C. Fuggle, G. A. Sawatzky, R. C. Karnatak, and J.-M. Esteva. 3d x-ray-absorption lines and the $3d^9 4f^{n+1}$ multiplets of the lanthanides. *Phys. Rev. B*, 32:5107, 1985.
- [39] E. Clementi, D. L. Raimondi, and W. P. Reinhardt. Atomic screening constants from scf functions. II. atoms with 37 to 86 electrons. *J. Chem. Phys.*, 47:1300, 1967.
- [40] J. C. Slater. Atomic radii in crystals. *J. Chem. Phys.*, 41:3199, 1964.
- [41] V. Anisimov and Y. Izyumov. *Electronic Structure of Strongly Correlated Materials*. Springer, 2010.
- [42] V. I. Anisimov, I. V. Solovyev, M. A. Korotin, M. T. Czyżyk, and G. A. Sawatzky. Density-functional theory and NiO photoemission spectra. *Phys. Rev. B*, 48:16929, 1993.

- [43] V. I. Anisimov, F. Aryasetiawan, and A. I. Lichtenstein. First-principles calculations of the electronic structure and spectra of strongly correlated systems: the LDA+ U method. *J. Phys. Condens. Matter*, 9:767, 1997.
- [44] E. Pavarini. The LDA+DMFT approach. In E. Pavarini, E. Koch, A. Lichtenstein, and D. Vollhardt, editors, *The LDA+DMFT approach to strongly correlated materials*, volume 1 of *Reihe Modeling and Simulation*. Forschungszentrum Jülich, 2011. Available: <http://hdl.handle.net/2128/4467>.
- [45] L. Vaugier, H. Jiang, and S. Biermann. Hubbard U and hund exchange J in transition metal oxides: Screening versus localization trends from constrained random phase approximation. *Phys. Rev. B*, 86:165105, 2012.
- [46] T. Fujiwara and M. Korotin. Spin and orbital ordering of $\text{Nd}_{1-x}\text{Sr}_x\text{MnO}_3$ from LSDA+ U calculations. *Phys. Rev. B*, 59:9903, 1999.
- [47] E. Koch. Mean-field theory: Hartree-Fock and BCS. In E. Pavarini, E. Koch, J. van den Brink, and G. Sawatzky, editors, *Quantum Materials: Experiments and Theory*, volume 6 of *Reihe Modeling and Simulation*. Forschungszentrum Jülich, 2016. Available: <http://hdl.handle.net/2128/12467>.
- [48] OEIS Foundation Inc. The on-line encyclopedia of integer sequences. <http://oeis.org>, 2019.
- [49] H. J. H. Clercx and P. P. J. M. Schram. An alternative expression for the addition theorems of spherical wave solutions of the helmholtz equation. *J. Math. Phys.*, 34:5292, 1993.
- [50] D. Sébilleau. On the computation of the integrated products of three spherical harmonics. *J. Phys. A*, 31:7157, 1998.
- [51] O. R. Cruzan. Translational addition theorems for spherical vector wave functions. *Quart. Appl. Math.*, 20:33, 1962.
- [52] A. Liberato de Brito. Fortran program for the integral of three spherical harmonics. *Comput. Phys. Commun.*, 25:81, 1982.
- [53] A. Liberato de Brito and T. P. Eggarter. Two useful identities involving spherical harmonics. *Int. J. Quantum Chem.*, 17:459, 1980.
- [54] F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, and B. V. Saunders, editors. NIST Digital Library of Mathematical Functions. <http://dlmf.nist.gov/>, Release 1.0.23 of 2019-06-15.
- [55] J. C. Slater and G. F. Koster. Simplified LCAO method for the periodic potential problem. *Phys. Rev.*, 94:1498, 1954.
- [56] W. A. Harrison. *Electronic structure and the properties of solids*. Dover Publications, 1989.

- [57] E. Koch. Quantum cluster methods. In E. Pavarini, E. Koch, D. Vollhardt, and A. Lichtenstein, editors, *DMFT at 25: Infinite Dimensions*, volume 4 of *Reihe Modeling and Simulation*. Forschungszentrum Jülich, 2013. Available: <http://hdl.handle.net/2128/7937>.
- [58] M. L. Boas. *Mathematical Methods in the Physical Sciences*. Wiley, 3rd edition, 2006.
- [59] R. D. Cowan. *The theory of atomic structure and spectra*. University of California Press, 1981.
- [60] G. Racah. Theory of complex spectra. I. *Phys. Rev.*, 61:186, 1942.
- [61] G. Racah. Theory of complex spectra. II. *Phys. Rev.*, 62:438, 1942.
- [62] G. Racah. Theory of complex spectra. IV. *Phys. Rev.*, 76:1352, 1949.

Eidesstattliche Erklärung

Ich, Qian Zhang,

erklärt hiermit, dass diese Dissertation und die darin dargelegten Inhalte die eigenen sind und selbstständig, als Ergebnis der eigenen originären Forschung, generiert wurden.

Hiermit erkläre ich an Eides statt

1. Diese Arbeit wurde vollständig oder größtenteils in der Phase als Doktorand dieser Fakultät und Universität angefertigt;
2. Sofern irgendein Bestandteil dieser Dissertation zuvor für einen akademischen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Institution verwendet wurde, wurde dies klar angezeigt;
3. Wenn immer andere eigene- oder Veröffentlichungen Dritter herangezogen wurden, wurden diese klar benannt;
4. Wenn aus anderen eigenen- oder Veröffentlichungen Dritter zitiert wurde, wurde stets die Quelle hierfür angegeben. Diese Dissertation ist vollständig meine eigene Arbeit, mit der Ausnahme solcher Zitate;
5. Alle wesentlichen Quellen von Unterstützung wurden benannt;
6. Wenn immer ein Teil dieser Dissertation auf der Zusammenarbeit mit anderen basiert, wurde von mir klar gekennzeichnet, was von anderen und was von mir selbst erarbeitet wurde;
7. Kein Teil dieser Arbeit wurde vor deren Einreichung veröffentlicht.

Datum

Unterschrift